用户手册



ID850QB 版本 3.20

集成调试器

操作

目标器件
V850 系列

文档号: U17964CA1V0UM00 (第1版)
出版日期: 2006年**4**月 CP(K)

[备忘录]

Windows是微软在美国及其他国家的注册商标或商标。

- **本文档信息先于产品的生产周期发布。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。**
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。本文件所登载内容的错误，本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的，对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息，应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失，本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性，但用户应同意并知晓，我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害（包括死亡）的危险，用户务必在其设计中采用必要的安全措施，如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为：

  “标准等级”、“专业等级”以及“特殊等级”三种质量等级。

  “特殊等级”仅适用于为特定用途而根据用户指定的质量保证程序所开发的日电电子产品。另外，各种日电电子产品的推荐用途取决于其质量等级，详见如下。用户在选用本公司的产品时，请事先确认产品的质量等级。

  “标准等级”： 计算机，办公自动化设备，通信设备，测试和测量设备，音频・视频设备，家电，加工机械以及产业用机器人。

  “专业等级”： 运输设备（汽车、火车、船舶等），交通用信号控制设备，防灾装置，防止犯罪装置，各种安全装置以及医疗设备（不包括专门为维持生命而设计的设备）。

  “特殊等级： 航空器械，宇航设备，海底中继设备，原子能控制系统，为了维持生命的医疗设备、用于维持生命的装置或系统等。

  除在本公司半导体产品的数据表或数据手册等资料中另有特别规定以外，本公司半导体产品的质量等级均为“标准等级”。如果用户希望在本公司设计意图以外使用本公司半导体产品，务必事先与本公司销售代表联系以确认本公司是否同意为该项应用提供支持。

（注）

（1） 本声明中的“本公司”是指日本电气电子株式会社（NEC Electronics Corporation）及其控股公司。

（2） 本声明中的“本公司产品”是指所有由日本电气电子株式会社或为日本电气电子株式会社（定义如上）开发或制造的产品。

M5 02.11-1

[备忘录]

前 言

读者对象	本手册适用于那些用V850系列设计和开发应用系统的用户
目的	本手册帮助用户理解ID850QB 的各种功能。
手册组织	本手册包括以下几部分:

概述
安装
启动和停止
与PM+联合使用
调试功能
窗口参考
命令参考

如何使用本手册

假定本手册的读者具有电学、逻辑电路以及微控制器、C语言、以及汇编语言的综合基础知识。

如欲了解V850系列等功能
请参考每个产品的硬件用户手册

如欲了解V850系列指令功能
请参考**V850ES体系结构用户手册(U15943E)** 或 **V850E1 体系结构用户手册 (U14559E)**.

约定

数据含义：	数据的高位部分在左边，低位部分在右边
注：	文中加有”注”的部分为脚注
注意事项：	需要特别注意的警告信息
备注:	附加信息
数字表示:	二进制… XXXX 或 XXXXB
	十进制... XXXX
	十六进制… 0xXXXX

前缀表示2的指数 (地址空间和内存容量):

K (Kilo): 2^{10} = 1024
M (Mega): 2^{20} = 1024^2

相关文档 与阅读本手册相关的文档如下。

在本文档中涉及的相关文档可能包含其初始版本，但初始版本并没有像这样标记。

和开发工具相关的文档(用户手册)

文档名称		文档编号
IE-V850E1-CD-NW(PCMCIA Card Type On-Chip Debug Emulator)		U16647E
QB-V850ESSX2(In-circuit emulator for V850ES/SG2, V850ES/SJ2)		U17091E
QB-V850IA4(In-circuit emulator for V850ES/IK1, V850E/IA3, V850E/IA4)		U17167E
QB-V850ESKX1(In-circuit emulator for V850ES/Kx1, V850ES/Kx1+)		U17214E
CA850 C语言编译器包Ver. 2.70	操作	U16932E
	C 语言	U16930E
	汇编语言	U16931E
	Link指令	U16933E
CA850 C语言编译器包 Ver. 3.00	操作	U17293E
	C语言	U17291E
	汇编语言	U17292E
	Link指令	U17294E
PM+项目管理器Ver.6.00		U17178E
ID850QB集成调试器Ver. 3.20	操作	本手册
SM+系统仿真器	操作	U18010E
	用户开放接口	U17663E
SM850 Ver. 2.50 系统仿真器	操作	U16218E
SM850 Ver. 2.00 或以上版本系统仿真器	外部用户开放接口规范	U14873E
RX850 Ver. 3.20 实时操作系统	基本	U13430E
	安装	U17419E
	技术	U13431E
	任务调试器	U17420E
RX850 Pro Ver. 3.20 实时操作系统	基本	U13773E
	安装	U17421E
	技术	U13772E
	任务调试器	U17422E
AZ850 Ver. 3.30 系统性能分析器		U17423E
PG-FP4闪存编程器		U15260E
TW850 Ver.2.00 性能分析调试工具		U17241E

目 录

图列表

图号　　　　　标题, Page

表格列表

表格编号　标题　页

第 1 章　概　述

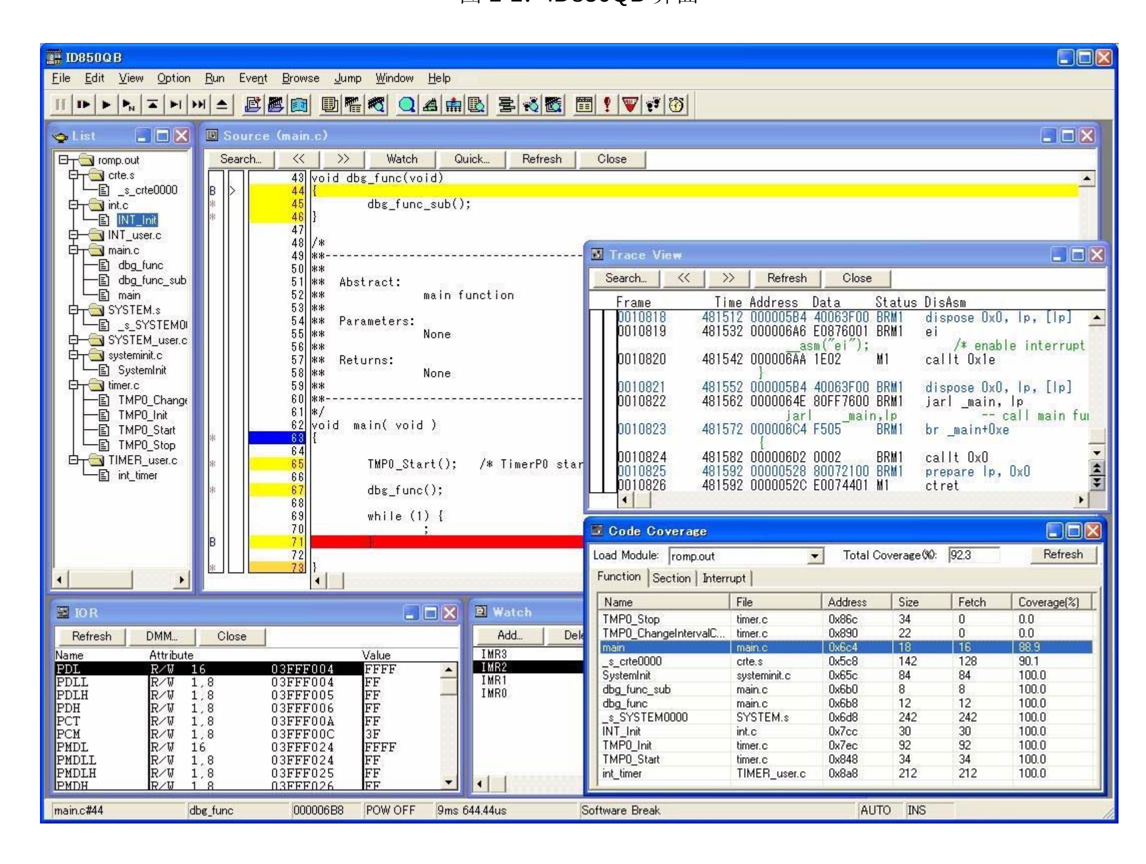
V850 系列集成调试器 ID850QB(以下称为 ID850QB)是为日电电子 V850 系列嵌入式微控制器开发的软件调试工具，其目的在于提高用户程序的调试效率。

ID850QB 与不同的仿真器连接可实现不同的功能

(括号中记述的符号即本手册中用于表示各调试设备的通用符号)

- 连接的设备为IECUBE™ (**[IECUBE]**)
- 连接的设备为N-wire调试卡或 MINICUBE™ (**[MINICUBE]**)
- 连接的设备为MINICUBE 2 (**[MINICUBE2]**)

图 1-1：ID850QB 界面



本章介绍ID850QB的基本特性

- 特征
- 系统配置
- 操作环境
- 调试中的注意事项
- GHS编译器的使用说明

1.1　特征

ID850QB具有以下特征:

- 新的功能，增强的功能
- 其他功能

1.1.1　新的功能、增强的功能

(1)　支持 MINICUBE2

支持市场上的小型片上调试仿真器 MINICUBE2。

(2)　支持 GHS 编译器(DWARF2 format)

支持 GHS 扩展的 DWARF2，其中 DWARF2 是和 Green Hill Software 公司编译器兼容的加载模块格式。(参考"1.5 使用 GHS 编译器的注意事项".)

(3)　增加了列表窗口

增加一个显示函数、变量、符号、节和中断请求的显示窗口 (参考" 列表窗口")。

可以通过不同的表单浏览信息，每一个项目都可以输出并保存为 CSV 格式。

(4)　支持非匹配数据格式

非匹配数据"Data Not Equal"是指事件与数据条件(读/写)不匹配的情况。当一个实例发生时非匹配数据被增加到状态条件中。(参考"事件对话窗"。)

(5)　支持数据闪存

支持从 V850ES/Fx3 的数据闪存中读写数据。

并且现在增加对 EEPROM 库错误仿真的支持。(参考"闪存操作对话框"。)

1.1.2　其他功能

(1)　使用在线(In-circuit)仿真器

当使用在线仿真器设置功能时，断点事件被设置，用户程序可以被跟踪，而且时间也可被测量。

(参考"5.12 事件功能"。)

(2)　支持片上调试器[MINICUBE]

通过Nx85ET(RCU0+TEU+TRCU)、Nx85E901(RCU0)，RCU1提供片上调试功能。

(3)　闪存写入功能[MINICUBE] [MINICUBE2]

可以通过通常的存储器操作对内部的闪存进行写入或下载加载模块(load module)操作。

(参考"闪存写入功能[MINICUBE][MINICUBE2]"。)

(4)　安全功能[MINICUBE] [MINICUBE2]

可以将ID码存储到内部ROM或内部闪存的安全单元中，并在每次进入调试之前进行认证。

(参考"配置对话框"中的(4) "ID码[MINICUBE][MINICUBE2]")。

(5)　通过 Tcl 实现功能扩展

批处理或钩子(hook)处理，以及创建原始用户窗口等功能都可以通过Tcl/Tk

(工具命令语言)的命令行操作实现。　(参考"第七章　参考命令"，"附录A　扩展窗口")

也支持最新的 Tcl/Tk 内核 8.4.12 和控制台功能。

(6)　通过 TIP 或 ToolLink 扩展功能

通过与支持任务调试器(RD) 和系统性能分析器(AZ)的工具接口协议(TIP)或 ToolLink 相关联，可以大幅提高对采用了实时 OS(RX) 的用户程序的调试效率。

1.2　系统配置

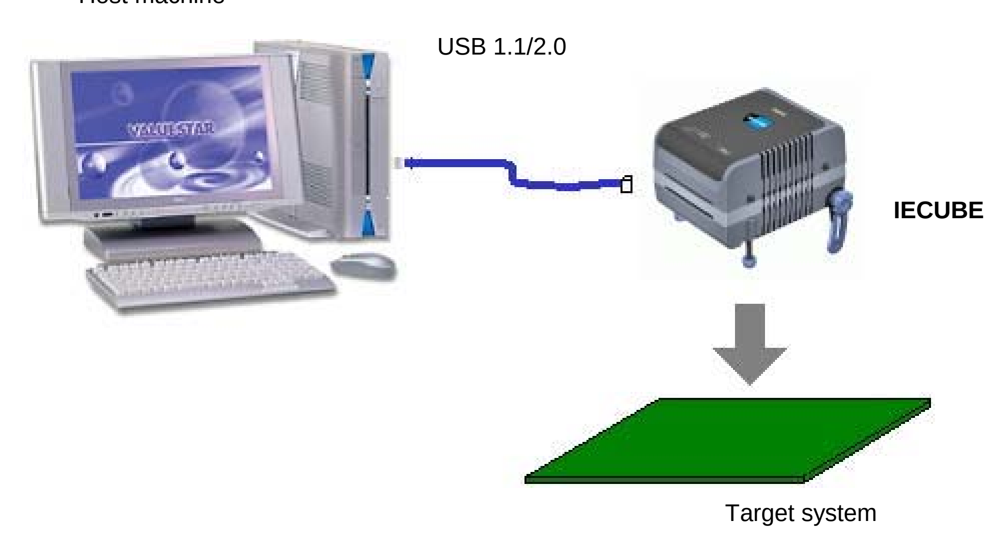
ID850QB可以与以下三种仿真器连接。

用户开发的 V850 系列的程序可以通过一个具有友好用户界面的进行调试。

(1)　IECUBE

把 IECUBE 和主机通过 USB 电缆连接后可以通过 ID850QB 对 IECUBE 进行操作。

图 1-2：ID850QB 系统配置范例(IECUBE)



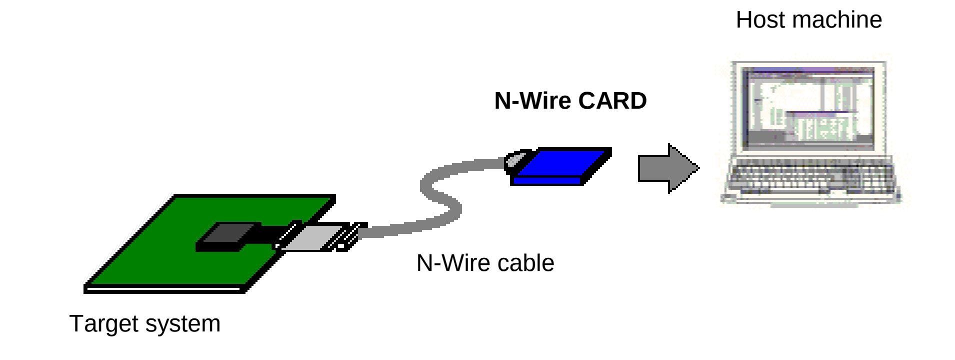
(2) N-Wire接口卡，MINICUBE

N-Wire 接口卡或 MINICUBE 可以对安装了配有运行控制单元(RCU)的 V850ES 器件或 V850E1 器件的目标系统提供调试功能。

把 **N-Wire** 接口卡直接插入 PC 机中便可以通过 ID850QB 对接口卡进行操作。

通过 USB 电缆把 MINICUBE 和主机连接便可以通过 ID850QB 对 MINICUBE 进行操作。

图 1-3：ID850QB 系统配置范例(N-Wire 卡，MINICUBE)





(3) MINICUBE2

通过 USB 电缆把 MINICUBE2 和主机进行连接，通过 ID850QB 实现对 MINICUBE2 的操作。

MINICUBE2通过和具有片上调试功能的微控制器连接提供调试功能。

图 1-4：ID850QB 系统配置范例(MINICUBE2)



1.3 操作环境

这一小节介绍系统的操作环境。

- 硬件环境
- 软件环境

1.3.1 硬件环境

(1) 主机(目标OS操作的PC机)

CPU	Pentium® II 400 MHz 或以上
内存	256 M字节或更多

注意事项: 当采用 N-**Wire** 接口卡连接时主机必须具有 PC 卡插槽(TYPEII)。

(2) 在线仿真器

- IECUBE (QB-V850Exxxx)
- N-Wire 接口卡(IE-V850E1-CD-NW)
- MINICUBE (QB-V850MINI)
- MINICUBE2 (QB-MINI2)

1.3.2 软件环境

(1) 操作系统(以下操作系统中任何一种)

Windows®98, Windows®2000, Windows®Me, Windows®XP(家庭版，专业版)

注意事项：无论采用以上的哪一种操作系统，我们推荐采用最新的软件补丁。

(2) 设备文件(单独下载)

- 将要使用的目标设备的设备文件。

备注: 这些文件可以从下面 NEC 电子公司的网站上下载：

http://www.necel.com/micro/index_e.html

(3)　支持工具(日电电子制造)

- CA850 C语言编译器(V3.10或更新版本)
- 工程管理器PM+(V6.10或更新版本)
- 系统性能分析器AZ850(V3.30或更新版本)
- 性能调节工具TW850(V2.10或更新版本)

1.4　调试中的注意事项

调试中的注意事项被描述如下。

- 当执行源代码级调试时的注意事项
- 安全ID的注意事项[MINICUBE][MINICUBE2]

1.4.1　当执行源代码级调试时的注意事项

目标源程序调试时必须包含符号信息或其它的调试信息 。

因此在源程序编译过程中需要执行以下操作。

(1)　当使用PM+时

当选择建立模式时指定[Debug Build]。

(2)　当不使用PM+时

指定-g选项。

1.4.2　安全 ID 的注意事项

当目标程序采用 N-WIRE 接口卡、MINICUBE 和 MINICUBE2 连接时必须包括安全 ID 信息(ID 代码)。

关于安全 ID 的设置，参考"CA850 操作"。

关于更详细的安全 ID 操作，参考 N-WIRE 接口卡、MINICUBE 和 MINICUBE2 的用户手册。

关于ID850QB的安全ID设置参考配置对话框。

1.5　关于 GHS 编译器的说明

ID850QB 的 V3.20 及以后的版本支持 Green Hills 软件公司的编译器，该编译器和 GHS 扩展的 DWARF2 格式兼容。

(1)　支持版本

- 集成开发环境MULTI ™(V3.51，V4.0.5)(DWARF2格式)
  注意事项：不支持C++，同时也不支持V850E2和V850E2R内核。

(2)　调试方式的选择项(调试方式)

- -G -dual_debug

第 2 章　系统安装

本章介绍下面 ID850QB 系统的安装方法 ：

- ID850QB的安装
- ID850QB的卸载

2.1　安装

当使用ID850QB，用户必须安装下列的内容。

表 2-1：安装

项目	程序
ID850QB	**当使用ID850QB盘安装的时候:** 使用自动执行的安装程序安装盘上内容。 **当使用从ODS网页下载的安装包安装时:** 运行下载的可执行文件，按照相关提示安装。
设备文件	用DFINST.exe单独启动安装程序，　按照[start] menu-> [All Programs] -> [NEC Electronics Tools] ->[DeviceFile Installer].步骤选择菜单项。

2.2　卸载

ID850QB的卸载可以通过控制面板中的[添加/删除程序]功能实现。

第 3 章　启动和停止

这一章介绍启动和停止ID850QB中的相关内容：

- 启动[MINICUBE] [MINICUBE2]前的注意事项
- 启动选项和参数指定
- 启动
- 停止
- 启动中的错误信息

3.1　启动前的注意事项

当连接 **N-Wire** 接口卡、[MINICUBE]或[MINICUBE2]时，在运行 ID850QB 之前要首先启动下列的检测工具，用来检测在线仿真器和目标系统能否被正常调试。

- **N-Wire**检查器**[MINICUBE]**
- OCD 检查器**[MINICUBE2]**

注意事项：有关片上仿真器和目标系统的连接方法以及上电次序请参考 N-**Wire** 接口卡，MINICUBE 和 MINICUBE2 的用户手册。不正确的连接可能导致在线仿真器和目标系统的损坏。

3.2　启动选项和参数设置

ID850QB 的启动选项和参数设置过程描述如下。通过指定启动选项和参数可以实现启动时对脚本文件和工程文件的指定。

备注: 当从PM+启动ID850QB时启动选项和参数设置是在PM+的[Tool]菜单中的[Debugger Setting …]中实现的(参考"第4章 与PM+联合使用")。在选项列中可以对调试器的启动选项进行设置。

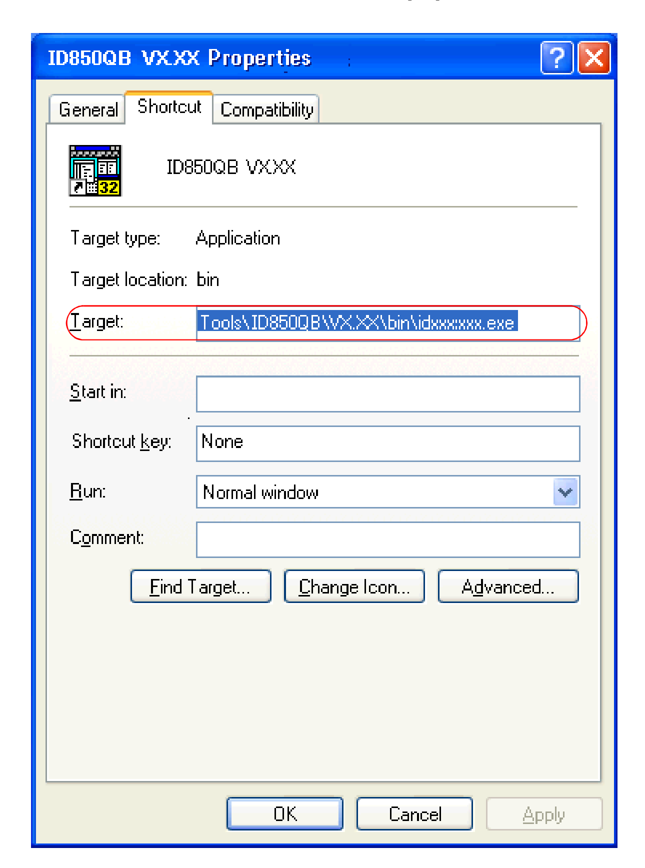
3.2.1　设置方法

1)　在桌面上创建ID850QB的快捷方式。

当执行完安装操作后，ID850QB 执行文件位于安装文件夹下的二进制文件夹中。

2)　打开创建快捷方式的属性，在[Target]中会显示可执行文件名称，指定选项和参数。(参考"3.3.2指定格式与选项"。)

图 3-1：启动选项 (例)



3.2.2　设置格式与选项

(1)　设置格式

```
id850g32.exe ?options?
id850g32.exe ?options? project
```

每一个选项和参数用空格分开。字符串中不区分大小写。

“?”之间的参数可以省略。

当指定了一个工程文件时，系统启动后首先读取该工程文件。

但是当 PM+启动后，工程文件的指定会被忽略。

当文件名和路径名字里面包含空格，工程文件和脚本文件应用双引号括起。(参考"例 3 路径名有空格时的设置")

(2)　设置选项

可以设置下列选项。

表 3-1：启动的选项

操作	意义
/sc	将窗口背景设为系统颜色。
/script:*script file name*	设置启动时执行的脚本文件。

(3) 设置的范例

例 1) 只指定脚本文件

```
id850g32.exe /script:c:\work\script.tcl
```

例2) 指定脚本文件和工程文件

```
id850g32.exe /script:c:\work\script.tcl c:\work\project.prj
```

例3) 路径名有空格时的设置

```
id850g32.exe /script:"c:\work folder\script.tcl" "c:\work folder\project.prj"
```

3.3 启动

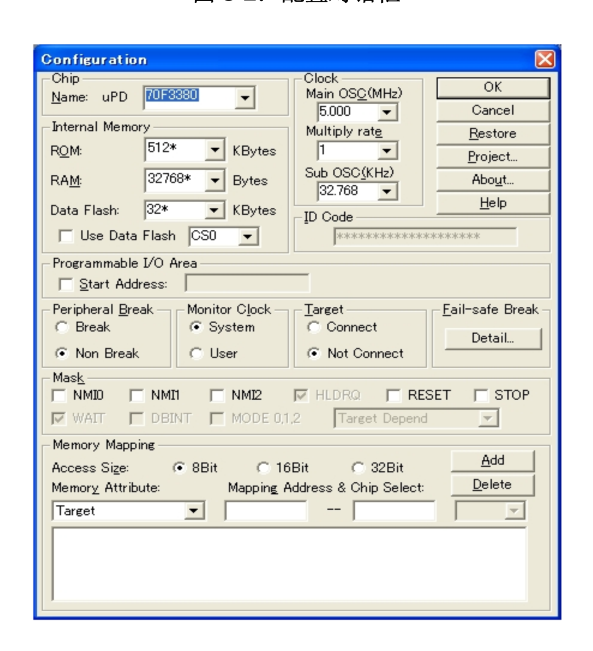
1) 可以从 PM+的[Start]菜单启动 ID850QB，或者从桌面上的快捷方式启动 ID850QB。

当从 PM+启动时请参考"4.3 从 PM+启动 ID850QB"。

当 ID850QB 启动时，配置对话框将被打开。

注意事项：在这种情况下，如果不显示配置对话框， 而是弹出一个错误信息，那么请参考 "3.5 启动过程中的错误信息"来处理。

图 3-2：配置对话框



2) 设置配置对话框中与ID850QB的操作环境相关的项。

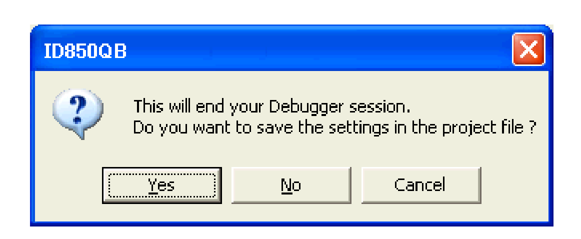
对每一项设置完后，单击对话框中的<OK>键。

3) 主窗口将被打开，用户可以操作ID850QB。调试工作是在主窗口中进行。

3.4　停止

1)　选择主窗口中[File]菜单中的[Exit]项，随后弹出一个退出调试对话框(当程序运行中执行停止操作时，系统显示一个执行停止消息)。

图 3-3：退出调试对话框



2)　保存当前的调试环境到一个工程文件，单击<Yes>按钮。如果单击<No>按钮，所有窗口将关闭，ID850QB 将停止。

3.5　启动中的错误消息

ID850QB 启动时的可能产生的错误消息列表如下(根据发生的顺序)。当这些消息产生时，请参考"附录 D 消息"。

注意事项：当连接多个仿真器时，各仿真器的优先级顺序如下：

IECUBE > MINICUBE > MINICUBE2 > N-Wire 调试卡

3.5.1　当连接 IECUBE 时

输出的错误消息如下表所示，消息的内容与目标系统的连接状态以及配置对话框中的设置有关。

表 3-2 错误信息输出类型　[IECUBE]

Error Message错误信息	设置对话框中的[目标]区域		目标		交换板		目标板电源	
	连接	未连接	连接	未连接	使用	未使用	打开	关闭
Ff606: 请检查与目标板的的连接，并且将其打开。	已检查							已检查
Wf607: 请检查与交换板的的连接。		已检查		已检查		已检查		已检查
Ff608: 请断开与目标板的的连接。		已检查	已检查					已检查
Ff609: 请关闭目标板的电源并断开与目标板的的连接。		已检查					已检查	

F0100: 无法与ICE通讯，请确认PC接口卡的驱动程序是否正确安装。
F0c43: 无法与仿真器连接。
F0c70: 无法与DCU连接。
F0c76: 与DCU的初始连接状态不正常。

F0c77: DCU访问不正常。
Ff606: 请检查与目标板的连接并将其打开。
Wf607: 请检查与转换适配器的连接。
Ff608: 请断开与目标板的连接。
A0105: 请断开与目标板的连接。
F0ca2: 此设备文件不包含片上调试信息。
F0ca4: 此设备文件不包含IECUBE信息。
F0c71: 无法复位。
F0c72: 无法访问监控程序区。
F0c73: 监控程序无法运行。
F0c74: 不可以访问CPU寄存器。
F0c23: 总线持续挂起。
A0c01: 访问寄存器时，CPU超时。
A0c02: 访问存储区时，CPU超时。
F0c04: 没有找到外部闪存数据库文件。
A01a0: Evachip没有反应，请确认CLOCK或者RESET WAIT，HLDRQ等等信号。 CPU没有反应，请确认CLOCK或者RESET WAIT，HLDRQ等等信号。

3.5.2 当连接 N-Wire 接口卡或 MINICUBE 时

F0100: 无法与ICE通讯，请确认PC接口卡的驱动程序是否正确安装。
F0c43: 无法与仿真器连接。
F03a0: 目标机没有打开。
F0c70: 无法与DCU连接。
F0c76: 与DCU的初始连接状态不正常。
F0c77: DCU访问不正常。
A0105: 读取设备文件失败(文件名)。
F0ca2: 此设备文件不包含片上调试信息。
F0ca3: 此设备文件中的片上调试信息包含不被支持的内容。
F0c24: 无法切换到调试模式
F0c72: 无法访问监控程序区。
F0c73: 监控程序无法运行。
F0c74: 不可以访问CPU寄存器。
F0c23: 总线持续挂起。
A0c01: 访问寄存器时，CPU超时。

A0c02: 访问存储区时，CPU超时。
F0c04: 没有找到外部闪存数据库文件。
A01a0: Evachip没有反应，请确认CLOCK或者RESET WAIT，HLDRQ等等信号。 CPU没有反应，请确认CLOCK或者RESET WAIT，HLDRQ等等信号。 .

3.5.3　当连接 MINICUBE2 时

F0100: 无法与ICE通讯，请确认PC接口卡的驱动程序是否正确安装。
F0c43: 无法与仿真器连接。
A01b2: 仿真器固件为旧版本，请更新为最新有效版本。
A0105: 读取设备文件失败(文件名)。
F03a0: 目标机没有打开。
F0ca2: 此设备文件不包含片上调试信息。
F0ca3: 此设备文件中的片上调试信息包含不被支持的内容。
F0ca1: 未找到监控程序。
F0c00: 监控程序访问错误。
F0c71: 无法复位。
F02a3: 持续复位。
F0c72: 无法访问监控程序区。
F0c24: 无法切换到调试模式。
F0c74: 不可以访问CPU寄存器。
F0c73: 监控程序无法运行。
F0c33 : 禁止关闭片上调试功能。
F0c34 : 禁止写入片上调试保留区域。
A010a: 不能同时运行调试器和一个实际应用。
A01a6: 执行器在运行。

第 4 章　PM+的关联操作

ID850QB 可以在开发过程中通过和 PM+的关联自动执行一系列操作，诸如创建源文件，编译，调试和修改源文件。

本章介绍以下和 PM+关联的操作事项。

关于 PM+更加详细的功能，请参考 PM+的用户手册。

- 设置编译模式。
- 在PM+工程中登录调试器
- 从PM+启动ID850QB
- 自动加载

注意事项: 如果加载模块文件是利用Windows的命令行创建的，就不能使用ID850QB和PM+关联的功能。

4.1　设置编译模式

为了在ID850QB中对用PM+创建的加载模块(load module)文件进行源代码级调试，在创建加载模块(load module)文件时必须编译输出调试信息。这可以通过设置PM+上的[Debug Build]来实现。

4.2　PM+工程中登录调试器

在 PM+中可以对每一个工程的调试器或加载模块(load module)文件进行指定。

4.2.1　选择调试器

选择调试器的过程如下：

通过下面的操作，ID850QB将成为激活工程的登录调试器，并且PM+的工具栏中会显示ID850QB的图标。

(1)　创建一个新的工作区(Workspace)

1)　选择PM+中[File]菜单中的[New Workspace]。

->这将打开一个创建新工作区的(workspace)向导对话框。

2)　使用向导对工作区进行必要的设置，[调试器设置]对话框将被打开，在对话框中指定ID850QB。关于置的细节请参考PM+的用户手册。

(2)　使用已经存在的工作区

1)　选择PM+中[工具]菜单中的[调试器设置]。

-> [调试器设置]对话框将被打开。

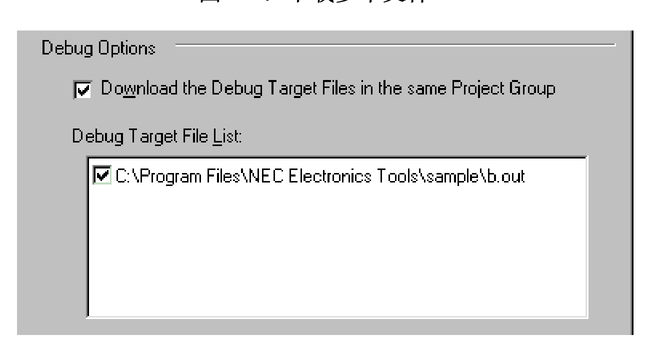
2)　指定ID850QB然后单击这个对话框中<OK>按钮。关于更详细的设置参考PM+的用户手册。

4.2.2　下载多个加载模块文件

当采用ID850QB调试器时，同一个工程组中可以下载多个加载模块文件。

下载多个加载模块文件是通过使用PM+中的"调试器设置"对话框来实现的。

图 4-1：下载多个文件



有关工程组更加详细的内容请参考PM+的用户手册。

备注:　可以在ID850QB的加载模块列表对话框中中对多个被下载的加载模块文件进行选择。

注意事项: 有内部闪存的产品，如果在[调试器设置]"对话框中选择"[加载后执行符号复位]"，在执行加载操作之前，内部闪存的内容将被删除**[MINICUBE]**

4.3　从 PM+启动 ID850QB

用户可以通过下列方式在PM+中启动ID850QB：

- 单击PM+工具栏中的ID850QB启动按钮。
- 选择PM+中[Build]菜单中[Debug]。
- 选择PM+中[Build]菜单中[Build and Debug]。
- 选择PM+中[Build]菜单中[Rebuild and Debug]。

如果ID850QB的调试环境被保存到一个PM+当前正在使用的工程文件中，启动时将调用此文件中所保存的调试环境设置。

如果ID850QB的调试环境没有被保存到一个PM+当前正在使用工程文件中，配置对话框将打开。在这种情况下，不允许改变设备类型(芯片名称)。

注意事项： 对于PM+，如果有太多的源文件被登录到一个工程中，文件数目可能超过ID850QB所允许登录的源路径长度的上限，这将导致源文件不能自动显示。

关于源路径长度的细节，参考调试器选项对话框中的"(1)源路径"。

4.3.1　恢复调试环境

当从PM+启动ID850QB时可以通过以下的过程恢复先前的调试环境：

1) 在PM+[注]中创建一个新的工作区(工程文件：例如，sample.prj)
2) 从PM+启动ID850QB。因为创建了新的工程文件，和单独启动ID850QB时的方式相同，

   在"配置对话框"中设置除了设备类型(芯片名称)以外的项目。
3) 通过ID850QB的下载对话框下载欲调试的加载模块(load module)文件。
4) 在ID850QB中调试加载模块文件。
5) 当ID850QB停止时，在"退出调试器对话框"中单击<Yes>按钮。

   ->调试环境将被保存至此工程文件(sample.prj)中(也可以在停止ID850QB前对工程文件sample.prj进行多次覆盖操作，这样也会将调试环境保存至文件sample.prj)。
6) 当下一次通过PM+启动ID850QB时，工程文件中保存的调试环境的信息将被自动恢复。

注： 在PM+和ID850QB中，调试环境信息被保存在工程文件中，可以被PM+和ID850QB使用的工程文件的扩展名是".prj"。工程文件中保存和恢复的信息细节参考相应产品的用户手册。

4.4　自动装载

如果用 ID850QB 调试加载模块文件时发生一个错误，可以通过下面的步骤修改源程序。文件的编译和重新下载可以自动执行(参考"4.4.1 通过修改源代码自动加载")。

当 ID850QB 处于激活状态，可以通过对 PM+中的文件进行编译和链接来重新下载加载模块至 ID850QB。(参考"4.4.2 通过启动调试器自动加载")。

注意事项：如果在PM+中选择了标准编辑器(idea-L)，则无法执行此过程。

4.4.1　通过修改源代码自动加载

修改源文件自动加载的过程如下：

1) 在源文件窗口中打开欲修改的源文件。选择[文件]菜单中[打开]，同时指定要在 ID850QB 修改的源文件

   (如果源窗口已经打开，这个窗口将被前置)。

   ->指定的文件将在源窗口内打开。

2) 在ID850QB中选择[编辑]菜单中的[编辑源程序]。

   ->打开一个编辑器并且读取指定的源文件。

3) 在编辑器中修改源文件。

4) 关闭编辑器。

注意事项：　在自动下载加载模块文件后，CPU 不复位。编辑器被调用时打开的调试窗口，每一个事件设置都将被恢复。如果在修改源文件时删除了一些语句或符号，将发生以下情况：

- 一个先前显示的变量变暗。
- 标记事件的标记被显示为黄色。.
- 软件断点可能被删除。

5) 在PM+中选择[Build]菜单中的[Build and Debug]项，或[Build]菜单中[Rebuild and Debug]项。

4.4.2　通过启动调试器自动加载

如果ID850QB启动时在PM+在中进行了下面的操作，加载模块将自动下载到ID850QB中。

- 选择PM+中的[Build]菜单中的[Build and Debug]项。
- 选择PM+中的[Build]菜单中的[Rebuild and Debug]项。

备注:　通过设定PM+中的[Tool]菜单中[Debugger Setting…]项来确定CPU是否在下载程序后复位。

(缺省情况下CPU将复位)

第 5 章　调试器功能

这一章介绍ID850QB调试器的功能。

表 5-1：调试器功能列表(调试操作的简要表述)

项目	参考
设置调试环境	5.1 设置调试环境
下载加载模块	5.2 下载功能，上传功能
显示源文件与反汇编结果	5.3 源文件与反汇编显示功能
设置一个断点	5.4 断点功能
执行用户程序	5.5 用户程序执行功能
检查变量值	5.6 窗口功能
检查并修改内存内容	5.7 内存编辑功能
检查并修改寄存器变量内容	5.8 寄存器操作功能
检查执行时间	5.9 定时器功能 [IECUBE]
检查被跟踪的数据	5.10 跟踪功能 [IECUBE]
检查代码覆盖率测量结果	5.11 代码覆盖率测算 [IECUBE]
对事件进行管理	5.12 事件功能
内存采样	5.13 RRM 功能
动态内存修改功能	5.14 动态内存修改功能
保存调试环境与窗口状态	5.15 加载/保存功能
跳转功能，联系窗口与告警消息	5.16 每个窗口都适用的通用功能

5.1　调试器环境设置

这一节介绍调试器环境的设置方法，具体内容包括：

- 设置操作环境
- 设置选项
- 设置映射
- 对控制访问外部存储器的寄存器进行修改

5.1.1　设置操作环境

ID850QB启动时配置对话框自动出现，在此对话框中对在线仿真器的操作环境进行设置。

如果一个工程文件已经存在，可以通过单击<Project…>按钮实现调试环境的重建(参考"5.15.1 调试环境(工程文件)"
)。

5.1.2　设置选项

在下列设置对话框中对调试器或在线仿真器进行相关设置。

- 配置对话框
- 扩展选项对话框
- 故障保险(Fail-safe)中断对话框
- RRM设置对话框
- 闪存选项对话框
- 数据闪存选项对话框
- 调试器选项对话框

5.1.3　映射设置

映射设置通过配置对话框进行。

表 5-2：映射属性

属性	含义
Emulation ROM **[IECUBE]** (带存储器板时)	仿真ROM 指定作为仿真ROM的存储区即目标设备连接后访问的ROM存储区。 目标设备访问在线仿真器中的存储区。 如果目标设备试图写入此存储区，将发生写保护断点(write protect break)。
Emulation RAM **[IECUBE]** (带存储器板时)	仿真RAM 指定作为仿真RAM的存储区即目标设备连接后访问的RAM存储区。 目标设备访问在线仿真器中的存储区。
Target	用户区映射 用户区映射指对目标板存储区进行访问的区域或CPU上集成的存储区。
Target ROM **[IECUBE]**	目标 ROM 指定作为目标的区域受到Fail-safe 断点保护。(参考" Fail-safe 断点对话框".)

属性	含义
I/O protect	I/O 保护区域 可以在指定的目标(Target)区域中指定I/O保护区域。I/O保护区域与不能在存储器窗口中显示的区域的显示方法相同(显示符号:??)。无法通过存储器窗口对这种属性的区域中的数据进行读写操作，因此，该区域也可以避免非法操作造成的错误读写。若要对映射在该区域的数据进行读写，则要将该数据登录到IOR窗口或观察窗口中。

5.1.4　对控制访问外部存储器的寄存器进行修改

当执行外部存储器映射时，下载之前必需先修改控制访问外部存储器的寄存器值，这可以利用IOR窗口或者钩子(hook) 程序实现。

如何利用钩子(hook)程序修改寄存器的值请参考"7.9 钩子(hook)程序"。

修改寄存器的值的方法，请参考CPU的硬件手册。

5.2　下载和上载功能

ID850QB允许下载和上载目标文件，目标文件格式列在表5-3和表5-4中。

本节介绍下列内容：

- 文件下载
- 文件上载

5.2.1 下载

在下载对话框中可以实现目标文件下载。

下载具有调试信息的加载模块文件时，相应的文本格式源文件会显示在源文件窗口中。

备注: 多个加载模块文件可以被下载。通过打开设置[File]菜单中的[Load Module]中的加载模块列表对话框可以选择已加载的文件。

可以被下载的文件格式如下：

表 5-3：可以被下载的文件格式

格式	扩展名
加载模块 (ELF/CA850 (.out)) 第三方加载模块(ELF/GHS 扩展 DWARF2 (.out))[注]	加载模块(*.out)
Intel十六进制格式(标准, 扩展, and extension linear)[Note2]	十六进制格式 (*.hex)
Motorola 十六进制格式S 类型	
扩展Tektronix 十六进制格式 (S0, S3, S7)	
二进制数据	二进制数据(*.bin)
覆盖率测试结果 **[IECUBE]**	覆盖率(*.cvb)

注1: 和GHS编译器V3.5.1和V4.0.x兼容

注2: 超过1MB的地址空间可以被下载(Intel的Hex格式)

备注: 自动确定的文件格式为*.hex。

5.2.2　上载

通过上载对话框可以实现存储器内容的上载，存储范围可以设置。可以上载的文件格式如下表所示：

表 5-4：可以被上载的文件格式

格式	扩展
Intel十六进制格式注1	十六进制格式(*.hex)Note2
Motorola 十六进制格式S 类型 (S0, S3, S7 - 32 位地址)	
扩展 Tektronix 十六进制格式	
二进制数据	二进制数据(*.bin)
覆盖率测试结果　**[IECUBE]**	覆盖率 (*.cvb)

注1:　　标准的16位地址，扩展的20位地址和扩展的32位线性地址。

超过1MB的地址空间可被上载(Intel的Hex格式)。

注2:　　当存储一个*.hex文件时可以选择列表中的一种格式。

5.3 源文件显示和反汇编显示功能

通过源文件窗口实现源文件显示，在汇编窗口中显示反汇编和在线汇编内容 。

这一节介绍如下内容

- 源文件显示
- 反汇编显示
- 混合显示模式(源窗口)
- 符号的转换(符号到地址的转换)

备注: 在源窗口和汇编窗口中会显示用户程序中已执行覆盖测量的位置。(参考"显示覆盖测量的执行位置")

5.3.1　源文件显示

下载包含调试信息的加载模块文件的同时，对应的文本文件在源文件窗口中显示。

在源文件移动对话框中可以修改显示的起始位置。这通过选择[View]菜单中的[Move…]项来实现。

在调试选项对话框中可以实现对制表符(tab)宽度、显示字体等以及源文件路径进行设置。 在源文件搜索对话框中点击

<Search…>按钮。搜索的结果在源文件窗口中以高亮方式显示。

表 5-5：可被显示的文件类型

文件类型(扩展名)	含义
源文件(*.c,*.s)	源文件(扩展名在调试选项对话框中可以修改)
文本文件(*.txt)	文本文件
所有文件(*.*)	所有文件

5.3.2　反汇编显示

通过汇编窗口可以实现反汇编显示。

通过打开[View]菜单中的[Move…]项可以在地址移动对话框中修改显示的起始地址。

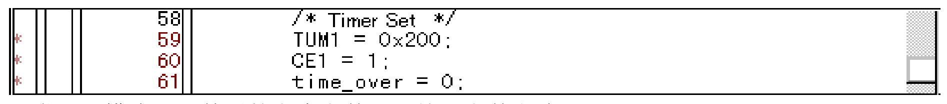
在调试选项对话框中可以指定显示偏移量和寄存器名称。

在汇编搜索对话框中点击<Search…>按钮可以指定搜索方式。搜索结果在汇编窗口中以高亮的方式显示。

5.3.3　混合显示模式(源窗口)

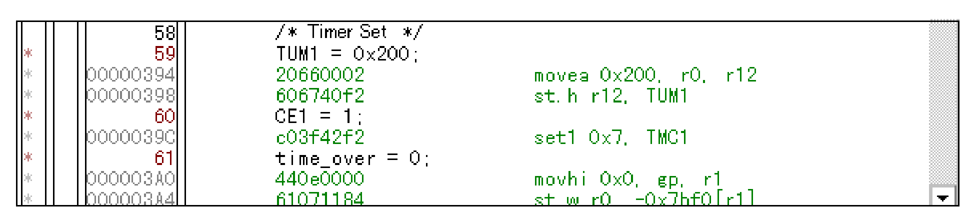
在源文件窗口中通过选择[View]菜单中的[Mix]选项可以实现源文件和反汇编文件的混合显示。混合显示的内容可以被存储为一个观察文件。

正常显示模式



正常显示模式下，普通的文本文件可以按源文件方式显示。

混合显示模式



如果程序代码与显示的某一行源文件相对应，反汇编代码会紧邻其下显示。地址标号、代码数据和反汇编指令都将被显示(指令的显示起始位置可以通过设置制表符的宽度来调整)。

注意事项:　只有当加载模块被下载，符号信息被读取且相应的源程序正在显示时,混合显示模式才是可用的。

备注:　在混合显示模式下使用光标键进行滚动操作时，可能产生过多的滚动。并且使用光标键滚动无法到达最后一行。

5.3.4　转换符号(符号到地址的转换)

在符号到地址转换对话框中，可以显示指定变量或函数的地址，或指定符号的值。

转换符号时,首先在源文件窗口或者汇编窗口中的选择欲转换的字符

串,然后选择关联菜单-> [Symbol...]。

指定符号的格式如下表所示

表 5-6：指定符号

转换目标	指定方法
变量	var file#var (用文件名指定一个静态变量) func#var (用函数名指定一个静态变量) file#func#var (用文件名和函数名指定一个静态变量)
函数	func file#func　(用文件名指定一个静态函数)
Label标号	label file#label (用文件名指定一个局部标号)
源文件行号	file#no prog$file#no
I/O 端口名称	portname端口名称
IOR 名称	I/O regname　I/O　Reg名称
Register name寄存器名称	Regname　Reg名称
PSW 标志名字	Pswname　psw名字

备注1:　　分隔符"#"

"#"用来分隔文件名、变量、函数名和行号。如果一个指定的符号在此范围内没有被找到，将对所有符号进行搜索(静态变量、动态函数和局部标号)。

备注2:　　分隔符"$"

当一次读取两个以上加载模块时，可以利用"$"作为分隔符来区别于文件名，变量，函数名，或符号名。

在缺省方式下，符号名优先级高。为了临时改变优先级，符号前加"$"的优先级高于寄存器名。

5.4　断点功能

断点功能用来停止 CPU 正在运行的用户程序或跟踪操作。

这一节介绍如下的内容：

- 断点类型
- 断点设置
- 对变量进行断点设置
- 硬件断点和软件断点
- 安全失败(Fail-safe)断点功能[IECUBE]

5.4.1 断点类型

ID850QB具有以下的断点类型。

表 5-7：断点类型

项目	内容
硬件断点注1 (事件触发的断点)	当满足断点事件条件时停止用户程序。 ->参考"断点设置"。
软件断点注1	用软件断点指令取代指定地址处的指令，同时停止用户程序的运行(参考"5.4.4硬件断点与软件断点")。 ->参考"5.4.2断点设置"
[运行到] 断点注2 (简单断点)	选中[Run]菜单-> [Come Here]项，运行到在源窗口或者汇编窗口中指定的地址时停止程序执行。
单步执行条件满足时跳出	当满足每个命令的停止条件时程序停止执行([Step In], [Next Over], [Return Out],[Slowmotion])。
强制断点	选择[Run]菜单中-> [Stop]或选择STOP按钮可以强制停止所有执行命令。 对所有的可执行命令有效
Fail-safe 断点**[IECUBE]**	当用户程序执行一个与存储区或者寄存器有关的非法操作时强制停止程序运行。(参考"5.4.5Fail-safe 断点功能[IECUBE]") ->参考" Fail-safe 断点对话框".
超时(time-out)断点**[IECUBE]**	测量的时间超过指定的时间长度(参考"5.9定时功能[IECUBE]") -> Refer to "参考"定时对话框".

注1: 对于[Go]，[Go & GO]，[Come Here]和[Restart]这种断点都是有效的

(参考"表5-9 执行类型"。)

注2: 用户程序停止执行后，这种断点被清除。在用户程序执行期间，光标位置前的断点事件不会发生。

5.4.2　断点设置

在源文件窗口或汇编窗口中的期望位置单击鼠标可以设置断点。

因为断点被设置为断点事件并且用事件功能来处理，因此可以设置断点的数目有限制(参考"5.12.4每个事件实例中可使用的事件数目")。

(1)　断点设置方法

单击有"*"显示的行设置断点 (程序代码存在的行)。

在缺省设置下，软件断点(B)被设置，但是如果在关联菜单中选择[Breakpoint]，则硬件断点(B或者B)被设置(参考"5.4.4硬件断点和软件断点")。

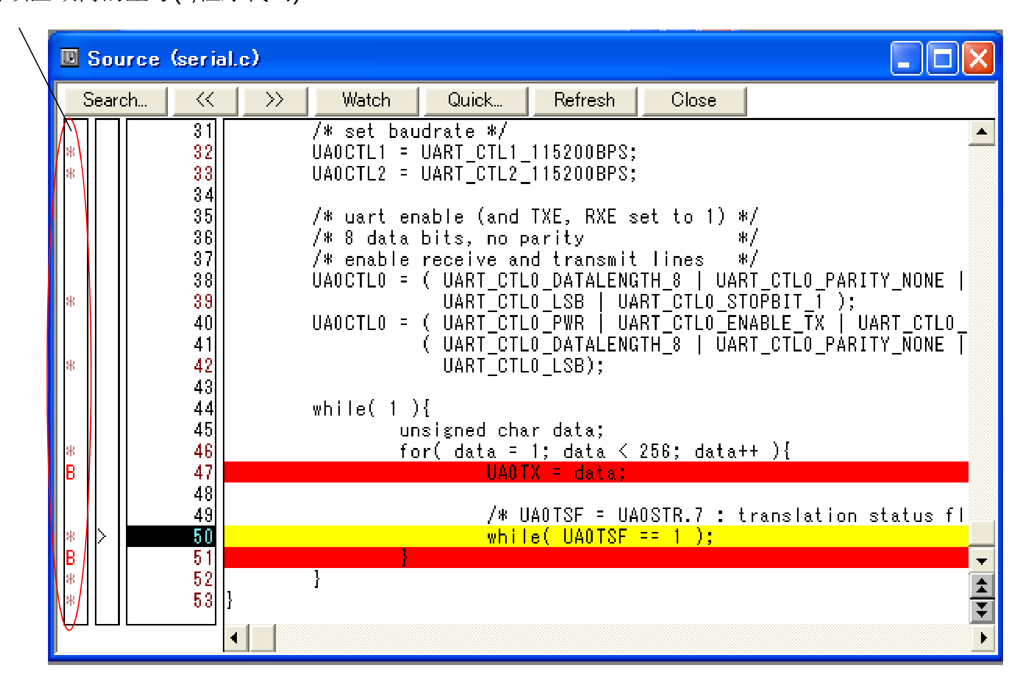
如果设置断点的行已经设置了事件断点，"A"标记出多个事件被设置(参考"表6-11事件设置状态(Event Mark)")。

注意事项:　在外部映射的ROM区域不能设置和删除软件断点。

备注:　可以通过扩展选项对话框修改缺省的断点设置。

图 5-1：断点设置

点击该区域内的星号(*;程序代码)



(2)　删除断点的方法

在设定断点的位置单击鼠标可以删除设置的断点。

与设置断点时一样，缺省情况下软件断点(B)被删除。但是如果在关联菜单中选择[Breakpoint]，

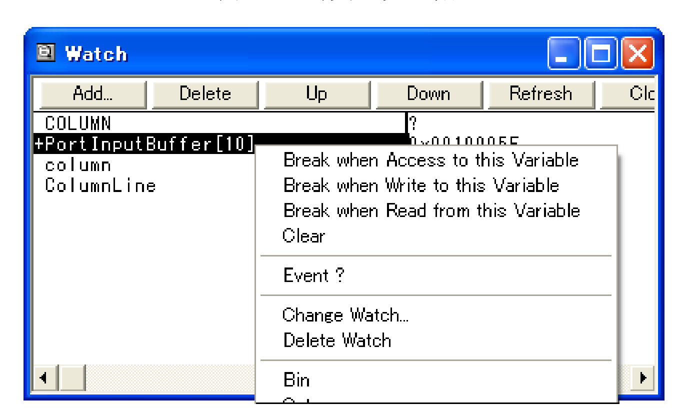
硬件断点(B和B)被删除。

执行删除后，如果另一个事件仍保留，保留事件的标记被显示。

5.4.3　对变量设置断点

在源窗口或观察窗口中的关联菜单中可以方便地设置访问断点。

图 5-2：对变量设置断点



5.4.4　硬件断点和软件断点

(1)　硬件断点

硬件断点指每个事件条件都使用了硬件资源的断点。

因此在ID850QB中，它们是被"5.12　事件功能"作为断点事件情况进行管理的。

有效的断点数目决定于使用的器件(参考"5.12.4每个事件实例中可使用的事件数目"。)

(2)　软件断点

软件断点是通过在指定的地址处重新写入软件断点指令来实现的。

在外部ROM和变量访问周期设置断点都是不允许的。

表 5-8：有效的软件断点数目

Connected IE	Valid Number
[IECUBE]	2000
[MINICUBE] **[MINICUBE2]**	2000[Note]

注:

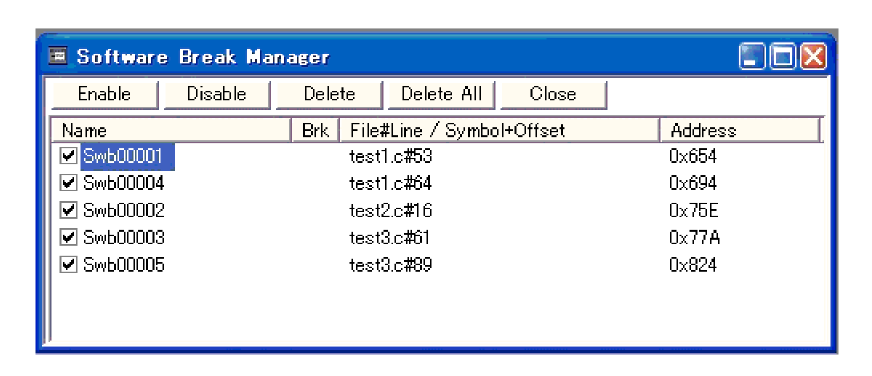
内部ROM和内部闪存相关的软件断点是通过ROM修正功能自动设置的。使用ROM修正功能可以设置的最大软件断点数目和产品有关(4)。而且通过ROM修正功能设置的软件断点在目标系统和内部系统复位期间被临时禁止，但是当一个断点事件发生时可以被激活。

注意事项:　当伪实时监控功能(读出时产生断点)被使能时，不能设置软件断点。

当一个有效的软件断点被设置后，用户程序执行期间的写操作被禁止(DMM Function).。 **[IECUBE]**

软件断点由软件断点管理器来管理。

图 5-3：软件断点的管理



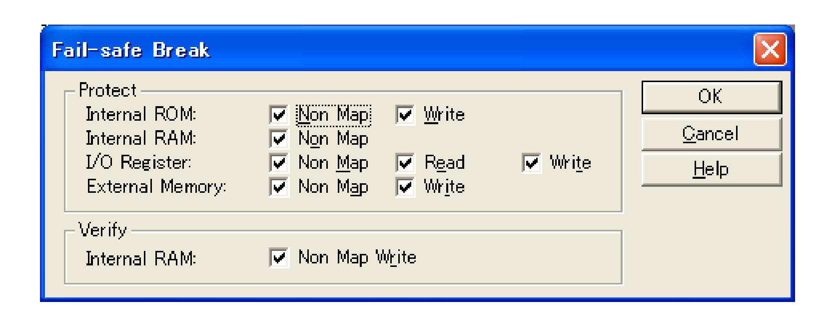
5.4.5　故障保险(Fail-safe)断点功能[IECUBE]+

故障保险断点的设置是在故障保险 (Faile-safe)断点对话框中进行的。通过选择复选框进行单独设置。

备注:　　　然而，内部RAM区的保护设置只有通过软件校验过程实现，因此在断点事件发生时时会出现一个警告。

在IOR合法性和内部RAM校验期间，状态栏中会显示地址。

图 5-4：安全失败(Fa-il-safe)断点设置



5.5　程序执行功能

程序执行功能用来启动和停止CPU执行或跟踪的用户程序。

用户程序的执行期间，程序计数器(PC)执行到断点或强制断点处停止(参考"5.4断点功能")。

备注:　程序执行期间，可以设置跟踪条件和定时器条件。(参考"跟踪对话框","定时器对话框") **[IECUBE]**

ID850QB提供如下所示的执行功能。这些功能的执行利用工具栏上的执行按钮实现，或者从[Run]菜单启动。

图 5-5：执行按钮

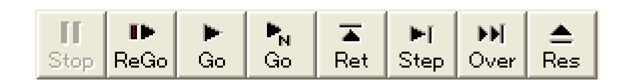


图 5-6：执行菜单

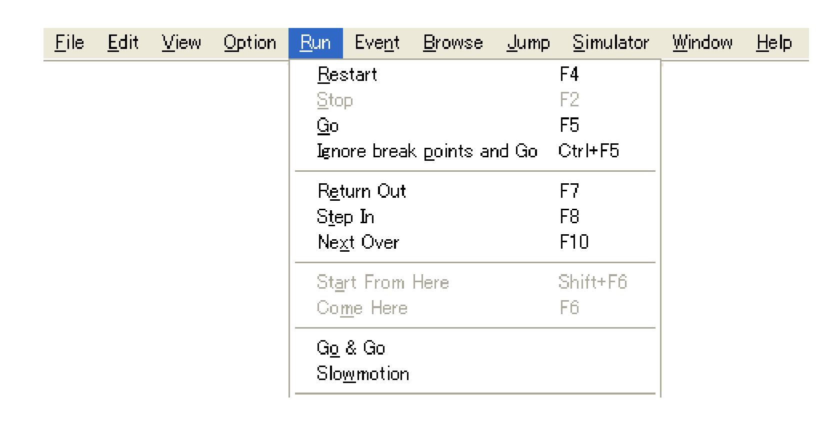


表 5-9：执行的类型

项目	内容
[Restart 复位]	CPU复位，用户程序从地址0开始执行。相当于"用户程序执行之前CPU复位并且执行[Go]"。
[Go运行]	用户程序从当前PC寄存器所指地址处开始执行，并且当所设断点条件满足时停止执行
[Ignore break points and Go忽略断点运行]	用户程序从当前PC寄存器所指地址处开始执行，忽略所设置断点。
[Return Out返回]	程序执行至返回到C语言调用函数为止。
[Step In单步执行]	在源文件模式下，从当前PC寄存器值所指地址处开始执行一行，同时更新各个窗口内容。 在汇编指令模式下，在当前PC寄存器值所指地址处执行一条指令，同时更新各个窗口内容。
[Next Over步越]	jarl指令： 执行下一步操作，被jarl指令调用的函数或子程序被视为一步(单步执行至jarl指令执行时的同一个嵌套层)。 Jarl之外的指令： 执行同步入[Step In]相同的操作。
[从当前位置开始执行] Start From Here	此命令从指定地址处开始执行程序。当一个设置的事件条件被满足时用户程序停止执行。
[Come Here运行到当前位置]	用户程序从当前PC寄存器值所指地址处开始执行，运行到源窗口或者汇编窗口显示区域所选中的当前行断点事件发生。当用户程序运行时，当前所设置的断点事件不发生。
[Go & Go]	用户程序从当前PC寄存器值所指地址处开始执行，设置的事件条件满足时停止。各个窗口内容被更新，程序从停止位置地址处继续运行。 此操作在用户[Stop]之前一直重复进行。
[Slowmotion]	在源文件模式下，用户程序从当前PC寄存器值所指地址处开始逐行执行。在指令模式下，指令逐条执行。各个窗口内容在每步执行完后更新。 此操作在用户[Stop]之前一直重复进行。
[CPU Reset　CPU复位]	CPU复位。
[Stop停止]	强制停止程序的运行。

5.6　观查功能

本节介绍以下与观察功能有关项:

- 数据值的显示和改变
- 局部变量的显示和改变
- 观察数据的登记和删除
- 改变观察数据
- 临时显示和改变数据值
- 调出观察功能
- 堆栈跟踪显示功能

5.6.1　显示和改变数据

在观察窗口中可以显示和改变数据的值。通过将观察数据登录至观察窗口可以检查数据值的改变。

调试选项对话框可以指定显示的方式

图 5-7：观察窗口

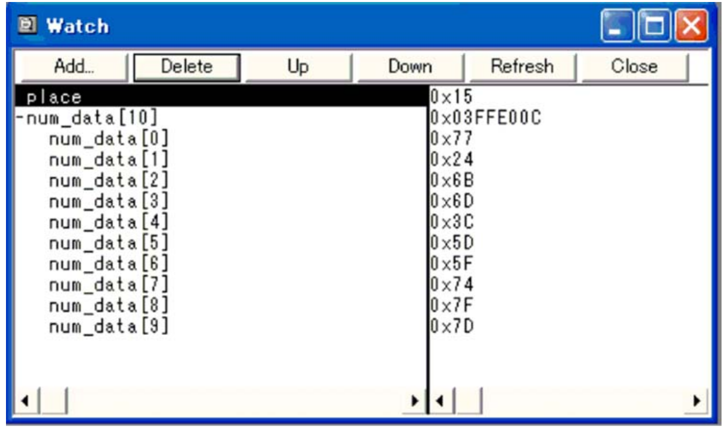
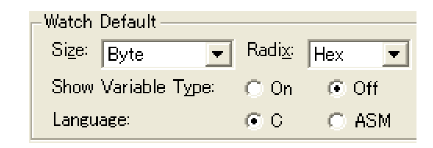


图 5-8：指定显示方式(调试选项对话框)

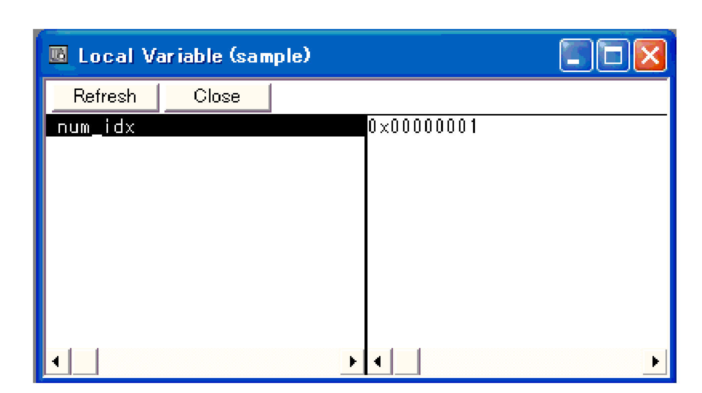


5.6.2　显示和改变局部变量的数值

在局部变量窗口中可以显示和改变局部变量的数值。

在这个窗口中当前函数使用的局部变量是自动显示的。 (不能增加和删除变量。)

图 5-9：局部变量窗口



5.6.3 登记和删除观察数据

用户可以从源窗口或汇编窗口登记数据到观察窗口。这可以通过在相应的窗口选择变量或符号名，然后单击<Watch”>按钮来实现。通过下列方法也可进行变量登记。

- 拖放选定的符号或变量到观察窗口(参考”5.16.4拖放功能”)。
- 在快速观察对话框或增加观察对话框中单击<Add>按钮。

为了删除观察数据，单击变量或符号名称(采用”Shift”和”Ctrl”键可以进行多选)，然后单击<Delete>按钮。但是，存在扩展层次内容的行，如数组的元素、结构或联合的成员是不能够被删除的。

5.6.4 改变观察数据

在改变观察对话框中可以改变所观察的数据。

即使改变后的符号名和已使用一个符号名称重复，这种符号名称的改变也是允许的。

图 5-10：改变观察对话框

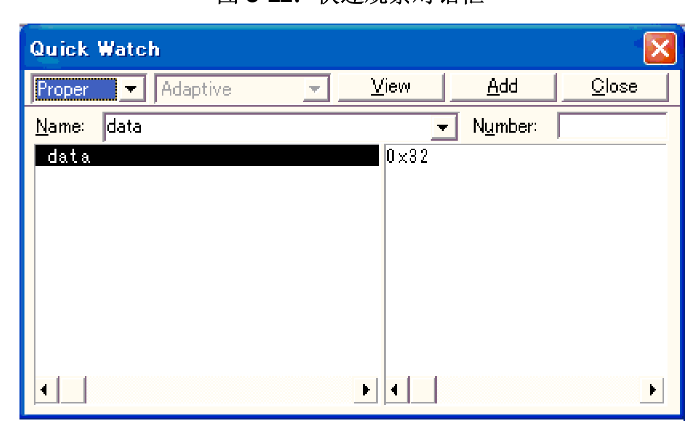
5.6.5　临时显示和改变数据值

在快速观察对话框中可以实现数据值的临时显示和改变。

在源窗口或汇编窗口中选择目标变量和符号，然后单击<Quick...>按钮执行观察数据登记。

在这个窗口中可以改变显示基点、窗口大小和显示数目。

图 5-11：快速观察对话框



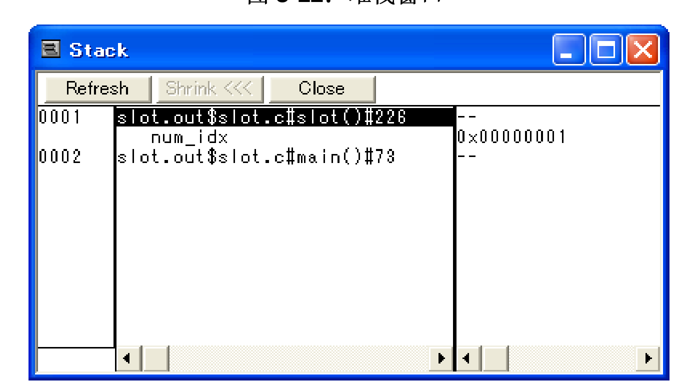
5.6.6　调出观察窗口

在源窗口和汇编窗口中当鼠标的光标放在选择的变量上时该变量值可以自动弹出。

5.6.7　堆栈跟踪显示功能

这个功能在堆栈窗口中显示当前用户程序的堆栈内容。

图 5-12：堆栈窗口



5.7　存储器操作功能

这一节介绍以下和存储器操作相关项：校验检查(Verify check)等等，详见扩展选项对话框。

- 显示和改变存储器的内容。
- 访问监视功能　[IECUBE]
- 存储器内容的填充、复制和比较
- 闪存写入功能[MINICUBE] [MINICUBE2]

5.7.1　显示和改变存储器内容

在存储器窗口中可以采用助记码、十六进制代码和ASCII码的形式对存储器内容进行显示和修改。在存储器搜索对话框中通过单击<Search>按钮可以实现搜索操作。搜索的结果在存储器窗口中以高亮的方式显示。

选择[View]菜单中[Move…]显示地址移动对话框，在此对话框中可以改变显示的起始地址。

即使在程序执行期间分配到采样区域的变量和数据的值也可以实时显示(参考”5.13中的RRM功能”)。

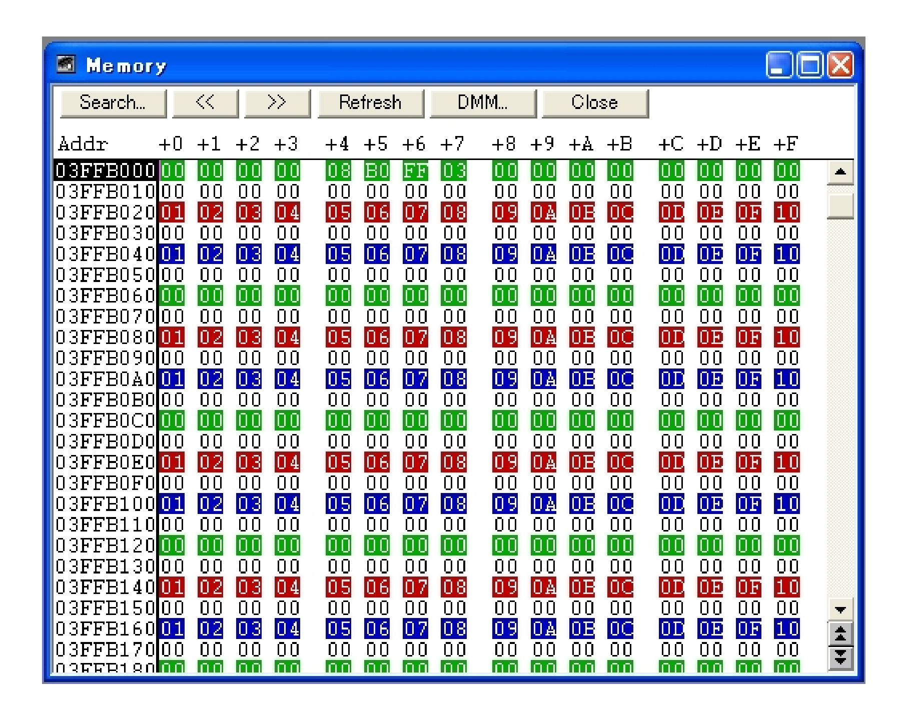
5.7.2　访问监视功能　[IECUBE]

在存储器窗口中RRM功能采样区可以实现访问监视功能，可以显示访问的状态(读状态，写状态，读写状态)。只有在选择以”字节”为显示单元时，才可以进行访问监视。ASCII码显示区没有颜色。通过选择[View] menu -> [Access Monitoring]以前的显示设置和访问状态显示可被清除。

注意事项1:　在程序执行期间通过DMA重新写入的存储器数值以及调试器重新写入的存储器数值不能在访问监视窗口中显示。

注意事项2:　只有在[Option]菜单中选择[RRM Function]后才能使用这个功能。

图 5-13：访问监视功能(存储器窗口)



5.7.3　存储器内容的填充、复制和比较

选择[Edit]菜单中[Memory]中的[Fill…/Copy…/Compare]后，出现存储器填充对话框，存储器复制对话框，和存储器比较对话框，在这几个对话框中实现存储器内容的填充、拷贝和比较。

比较结果显示在存储器比较结果对话框。

5.7.4　闪存写入功能[MINICUBE] [MINICUBE2]

在ID850QB中，内部闪存可以象访问普通存储器那样写入数据和下载程序加载模块。

通过存储器窗口，汇编窗口，观察窗口，存储器填充对话框和存储器拷贝对话框可以实现对内部闪存数据的更改，用户不必知道数据是否在内部闪存。通过闪存自身的编程功能也可以实现向内部闪存下载加载模块(参考"闪存选项对话框"和"数据闪存选项对话框")。

注意事项:　在程序执行期间，数据不可写入内部闪存。

备注:　在ID850QB中，在加载模块下载到内部闪存后其余区域的数据内容会被删除。

5.8　寄存器操作功能

这一节介绍下面的寄存器操作功能：

- 显示和改变寄存器内容的操作
- 显示和改变外围I/O寄存器的内容
- 显示和改变I/O端口的内容

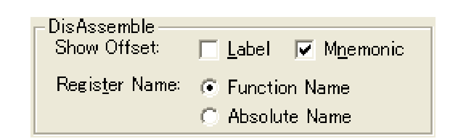
5.8.1 显示和改变寄存器的内容

在寄存器窗口可以显示和改变寄存器的内容。

在调试选项对话框中可以进行寄存器名称(绝对名称/功能名称)的切换操作。

注释: 在寄存器选择对话框中选择需要显示的寄存器。

图 5-14：绝对名称/功能名称的切换



5.8.2 显示和改变周边 I/O 寄存器的内容

在IOR窗口中可以显示和修改外围I/O寄存器的内容。

选择[View]菜单中[Move…]，地址移动对话框出现，在此对话框中可以改变显示的起始地址。

在IOR选择对话框中选择要显示的寄存器。

图 5-15：显示 IOR 的内容

IOR

Refresh DMM... Close

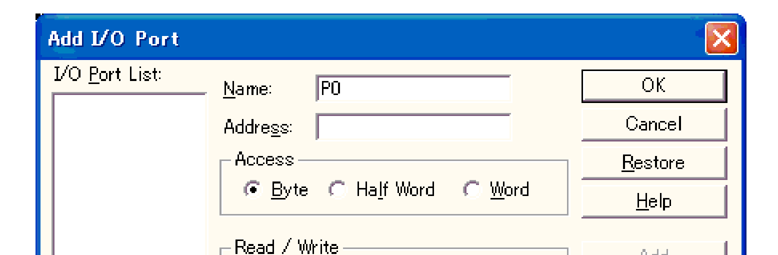
Name	Attribute			Value
PDL	R/W	16	03FFF004	0000
PDLL	R/W	1,8	03FFF004	00
PDLH	R/W	1,8	03FFF005	00
PDH	R/W	1,8	03FFF006	00
PCS	R/W	1,8	03FFF008	00
PCT	R/W	1,8	03FFF00A	00
PCM	R/W	1,8	03FFF00C	00
PCD	R/W	1,8	03FFF00E	00
PMDL	R/W	16	03FFF024	FFFF
PMDLL	R/W	1,8	03FFF024	FF
PMDLH	R/W	1,8	03FFF025	FF
PMDH	R/W	1,8	03FFF026	FF
PMCS	R/W	1,8	03FFF028	FF
PMCT	R/W	1,8	03FFF02A	FF
PMCM	R/W	1,8	03FFF02C	FF
PMCD	R/W	1,8	03FFF02E	FF
PMCDL	R/W	16	03FFF044	0000
PMCDLL	R/W	1,8	03FFF044	00
PMCDLH	R/W	1,8	03FFF045	00
PMCDH	R/W	1,8	03FFF046	00
PMCCS	R/W	1,8	03FFF048	00
PMCCT	R/W	1,8	03FFF04A	00
PMCCM	R/W	1,8	03FFF04C	00
BSC	R/W	16	03FFF066	5555
VSWC	R/W	8	03FFF06E	77

5.8.3　显示和改变 I/O 端口的内容

如果用户定义的I/O端口一旦在增加I/O端口对话框中登录，在IOR窗口中就可以对用户定义的I/O端口进行显示和修改。

如果产品支持可编程的I/O端口寄存器，在配置对话框中可以显示和修改可编程的I/O端口寄存器的内容。

图 5-16：寄存器 I/O 窗口



5.9　定时器功能[IECUBE]

定时器功能是用来测量用户程序从执行开始到一个断点的时间(运行-断点时间)，或者使用定时器事件测量用户指定的一段程序代码执行的时间。

ID850QB定时器功能通过外部时钟实现测量。因此可测量的时间基于"表6-7定时计数器分频比和最大测量时间的关系(定时计数器(定时器))"的设置而变化。

在主窗口的状态栏中可以显示运行-终止时间。

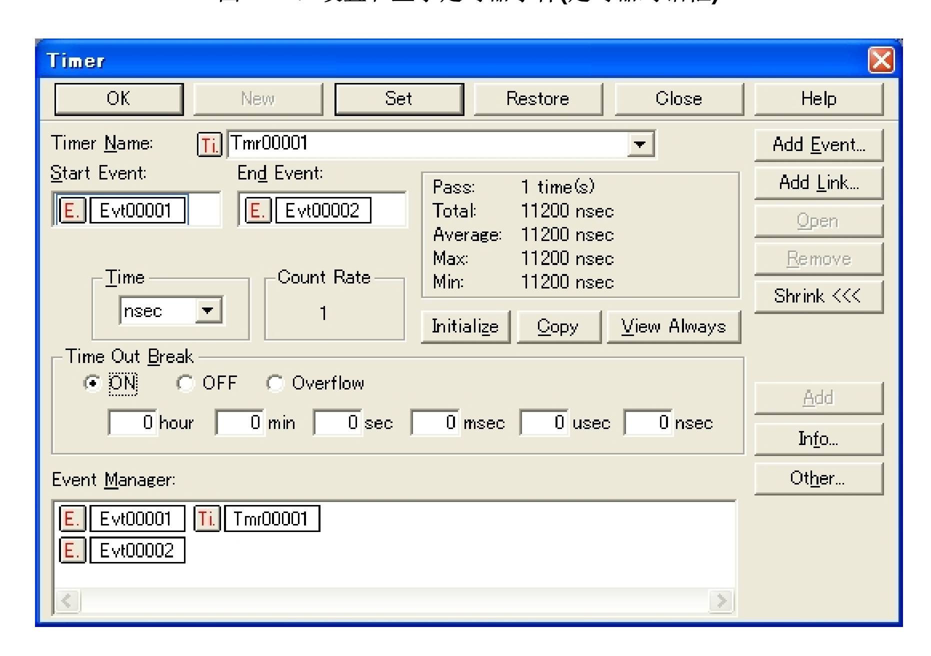
这一节介绍以下的功能：

- 定时器事件条件
- 运行-断点事件

5.9.1　定时器事件条件

定时器事件条件指定时间测量的启始和停止条件。在定时器对话框中设置定时器事件条件(参考"5.12事件功能")。在ID850QB中，在超时断点区域可以进行超时断点设置。

图 5-17：设置和显示定时器事件(定时器对话框)



通过单击<View Always>按钮，在定时器结果对话框中可以实现连续显示。

通过设置[Run]菜单中[Timer Start/Timer Stop]功能可以在程序执行期间实现定时器的操作。

5.9.2　运行-断点事件

运行-断点事件是测量从开始执行到断点的运行时间的定时器事件的名称。运行-断点事件事先设置。运行-断点时间在定时器对话框中指定时可以看到。

运行-断点时间也可以在主窗口的状态栏中显示。

因为运行-断点事件属于可以同时被使能的定时器事件之一(参考"5.12.4对每个事件条件使能事件的数目")，可以把它们加到有效的定时器事件条件中。

5.10 跟踪功能[IECUBE]

跟踪功能用来将指示用户程序运行过程的历史数据保存到跟踪存储器中。

无论跟踪条件如何DMA(直接存储器访问)的起始点和终止点都是被跟踪的(参考"5.10.7 DMA传输点跟踪功能")。

这一节解释如下内容：

- 跟踪存储器
- 设置跟踪数据
- 检查跟踪数据
- 混合显示模式(跟踪显示窗口)
- 跟踪操作
- 设置条件跟踪
- DMA传输点跟踪功能

注意事项： RRM功能、跟踪功能[IECUBE]和覆盖率测量功能[IECUBE]是互斥的(参考"5.11.4 互斥的RRM功能，跟踪功能和覆盖率测量功能")。

通过[Option]菜单可实现三种功能的切换。

5.10.1　跟踪存储器

ID850QB跟踪存储器是环形的。其大小的指定是在 扩展选项对话框 实现的。最大跟踪存储器容量如下所示：

表 5-10：跟踪存储器大小

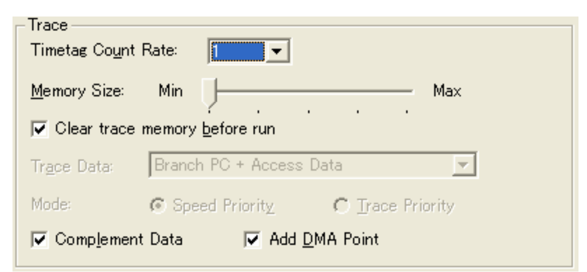
与IE连接	最大值
[IECUBE]	256 KB

5.10.2　设置跟踪数据

用户可以在"扩展选项对话框"中实现收集到的跟踪数据的详细设置。

在转移指令之间补充显示的指令不能被硬件跟踪，但在补充(Complement)模式(通过选中Complement Data area 复选框使能)下可以被跟踪。当补充模式被使能后，程序执行期间可以对内部ROM区进行汇编显示(当跟踪停止时)。

图 5-18：设置跟踪数据



5.10.3　检查跟踪数据

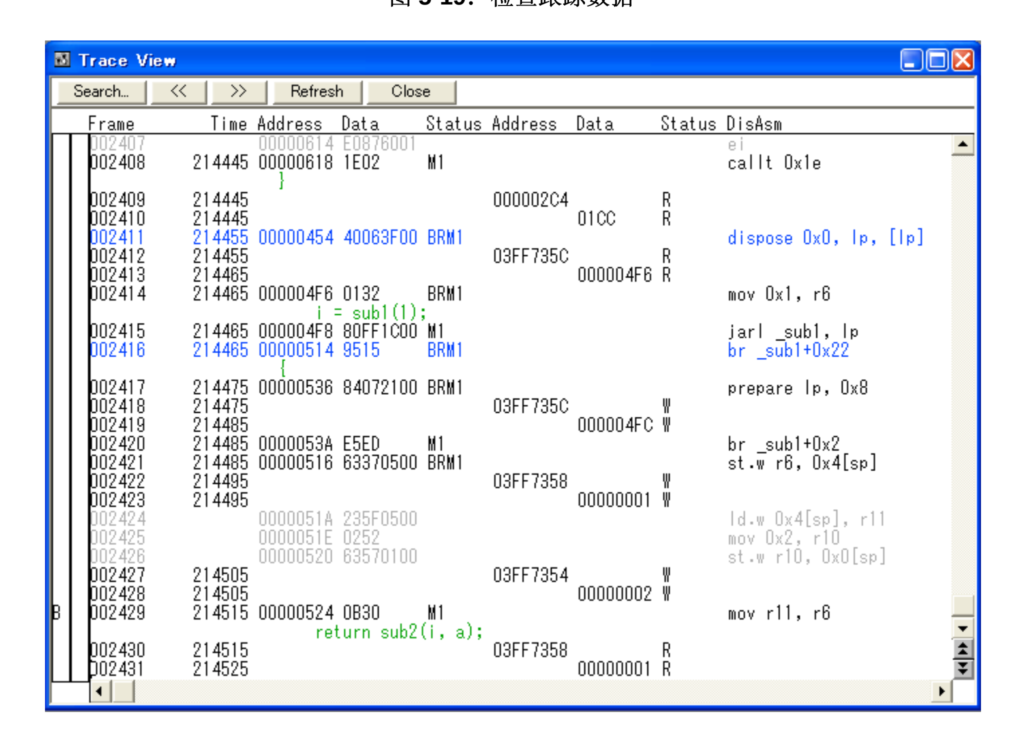
存储在跟踪存储器中的数据可以在跟踪观察窗口中进行检查。用户在跟踪搜索对话框中单击<Search>按钮可以搜索跟踪的数据。

通过选择[View]菜单中[Move]打开地址移动对话框可以改变显示的起始地址。

在跟踪数据选择对话框中可以选择跟踪观察窗口内的显示项目。

在跟踪观察窗口可以设置计数器的分频比，跟踪观察窗口中的时间标记显示用到此数值。分频比是在扩展选项对话框中设置的。时间标签计数器、计数速率以及全局时间标签是在扩展选项对话框中实现的。

图 5-19：检查跟踪数据



5.10.4　混合显示模式(跟踪观察窗口)

在跟踪观察窗中选择[View]->[Mix]可以实现跟踪结果和源文件的混合显示(混合显示模式)。

如果代码恰好处在程序取地址的位置，源文件行在跟踪取指地址行的前一行显示。

源文件行用绿色突出显示。

```
Frame   Time Address  Data      Status Address  Data     Status          DisAsm
32757      3 000005A2 85058505 BRM1                                      br _main+0x21a
                     while(1);
32758      3 000005A2 85058505 BRM1                                      br _main+0x21a
                     while(1);
32759      3 000005A2 85058505 BRM1                                      br _main+0x21a
                     while(1);
32760      3 000005A2 85058505 BRM1                                      br _main+0x21a
                     while(1);
```

注意事项： 只有所包含的符号信息已经被读取，加载模块已被下载，取指地址，取指数据，取指状态或者反汇编的结果被显示时，混合模式才是有效的。

5.10.5　跟踪操作

跟踪操作会因用户程序执行模式和跟踪控制模式的不同而有所不同。

通过选择[Run]->[Tracer Start/Tracer Stop]来实现程序运行期间的跟踪操作。

(1) 在执行期间操作

跟踪操作会因菜单[Run]->[Cond. Trace ON/Uncond. Trace ON]选择的不同而不同。

表 5-11：跟踪模式的类型

项目	内容
无条件跟踪	用户程序执行时跟踪开始，断点事件发生时结束。 此时跟踪事件触发条件的设置被忽略。
条件跟踪	根据跟踪对话框中的条件设置开始与结束跟踪。(参考 “5.10.6设置条件跟踪”) 跟踪过程中如遇断点，跟踪立即结束。

(2)　在单步执行(Step In)时的操作

跟踪每一步的执行，每一次跟踪的数据被成功增加到跟踪存储器中。

(3)　在步越(Next Over)执行时的操作

这种跟踪方式因步越(Next Over)的指令不同而不同。

(a) jarl disp22, [lp] 指令

jarl指令和被调用的子程序被跟踪。

(b)其他指令

与单步执行(Step In)时执行的操作相同。

(4)　跟踪控制模式

跟踪控制模式有以下的类型。通过[Run]菜单可以实现跟踪模式的设置。

表 5-12：跟踪控制模式的类型

模式	内容
不停止(Non Stop)	循环跟踪跟踪内存并覆盖最旧的页面(默认情况)。
完全停止(Full Stop)	跟踪所有的跟踪内存然后停止跟踪。
完全断点(Full Break)	跟踪所有的跟踪内存然后停止跟踪与程序执行
延时触发停止(Delay Trigger Stop)	根据延时计数帧的数目跟踪数据，当延时触发事件发生时停止跟踪。

5.10.6　设置有条件的跟踪

当设置条件跟踪时，跟踪条件满足时可以触发跟踪开始/结束操作。

通过在跟踪对话框中设置跟踪发生的条件，来设置有条件的跟踪操作。(参考"5.12事件功能")

通过选择[Run]菜单中[Cond. Trace ON]来使用有条件的跟踪。

有条件的跟踪的类型如下所示

表 5-13：有条件跟踪的类型

项目	内容，设置方式
区域跟踪(Section trace)	在两个指定的条件之间跟踪(在一个指定的区域)。通过在跟踪对话框内设置一个跟踪开始事件和一个跟踪结束条件并选择[Run] -> [Cond. Trace ON]来执行区域跟踪
限定跟踪(Qualify trace)	仅当跟踪条件被满足时执行跟踪。 如果设置了两个以上事件作为条件跟踪事件，可以通过运行条件跟踪来实现限定跟踪。 可以通过在跟踪对话框中选择[Run] -> [Cond. Trace ON]设置一个限定跟踪事件来执行限定跟踪。
延时触发跟踪(Delay trigger trace)	在某个条件被满足后，根据延时计数值进行跟踪。 通过在跟踪对话框中设置延时触发事件来进行延时触发跟踪，在延时计数对话框中设置一个延时数值并且选择[Run] -> [Cond. Trace ON]

5.10.7　DMA 传输点跟踪功能

DMA传输点跟踪(Direct Memory Access Trace)功能具有比普通跟踪功能高的优先级。

对于DMA传输点跟踪功能访问的帧，在跟踪观察窗口中标记"M"。

下面是DMA传输点跟踪功能的类型。

(1)　无条件跟踪期间

不但所有的采用正常全跟踪功能跟踪的点，而且DMA起始点和终止点总是被DMA传输点跟踪功能所跟踪。

(2)　区域跟踪期间

DMA传输点跟踪功能不但可以跟踪正常段跟踪功能跟踪的点，而且可以跟踪DMA的起始和终止点，这些点可以位于区域跟踪区域以外。

(3)　在限定的跟踪期间

DMA传输点跟踪功能不但可以跟踪正常限定跟踪功能跟踪的点，而且可以跟踪DMA的起始和终止点，这些跟踪点甚至可以不满足限定跟踪的条件。

5.11　覆盖率测量功能[IECUBE]

覆盖率测量的方法有很多种，ID850QB中采用了C0覆盖率测量法来进行代码覆盖率的测量。

C0覆盖率(指令覆盖率)：测量至少执行一次的代码所占的百分比。

通过下载对话框和上传对话框可以分别下载和上传覆盖测量结果(coverage data)。

这一节介绍下列内容：

- 覆盖率测量结果显示
- 覆盖率测量范围
- 对已进行覆盖率测量部位的显示
- RRM功能，跟踪功能，和覆盖率功能是互斥的。

注意事项**1:**　如果没有覆盖板，这个功能不能实现。

注意事项**2:**　RRM功能，跟踪功能[IECUBE]，和覆盖率测量功能[IECUBE]是互斥的(参考"5.11.4在一个独占的交互环境下的RRM、跟踪和覆盖测量功能")。

在[Option]菜单中可实现三种功能之间切换。

5.11.1　覆盖率测量结果显示

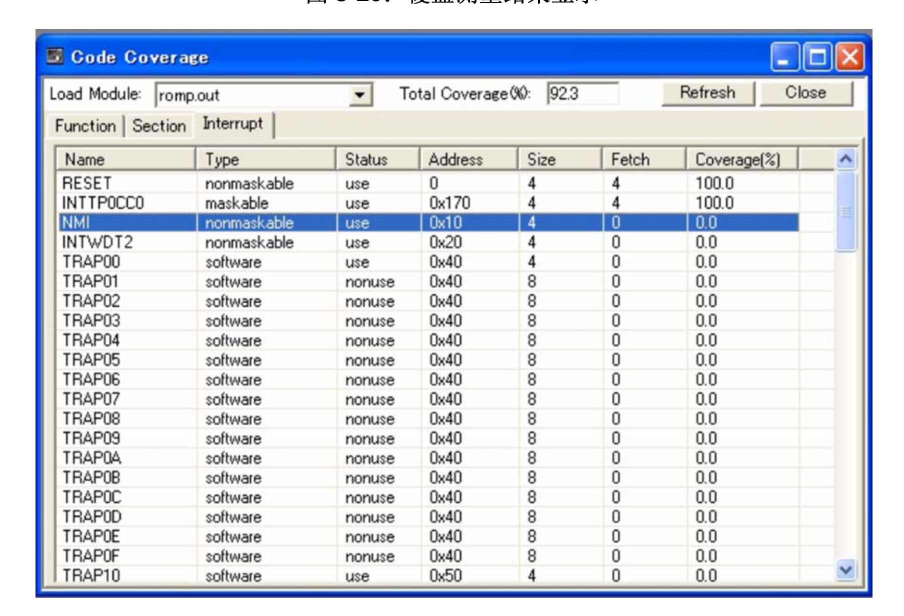
可以在代码覆盖率窗口中检查覆盖率测量结果

在代码覆盖率窗口中，对每个函数、段和中断句柄(vectors)单独显示覆盖率测量结果。在一个断点处覆盖率测量结果将被更新(但是在程序执行期间不会自动更新)。通过选择[Option]菜单可以清除覆盖率数据。覆盖率数据可以被存储为CSV格式(参考"5.15.2窗口显示信息(查看文件)")。

存储内容会因制表栏中选择不同而不同。

通过选择[Option]菜单中[Coverage->Clear]来清除覆盖数据。

图 5-20：覆盖测量结果显示



5.11.2　覆盖率测量范围

覆盖率测量范围如下所示。

表 5-14：代码覆盖率测量范围

连接IE	代码覆盖率测量范围
[IECUBE]	内部ROM空间 ＋ 任何 1 MB空间(在覆盖-地址对话框中可选择的空间)

5.11.3　对已进行覆盖率测量部位的显示

对于已执行覆盖率测量的部位会在源文件窗口和汇编窗口中基于测量信息进行显示。

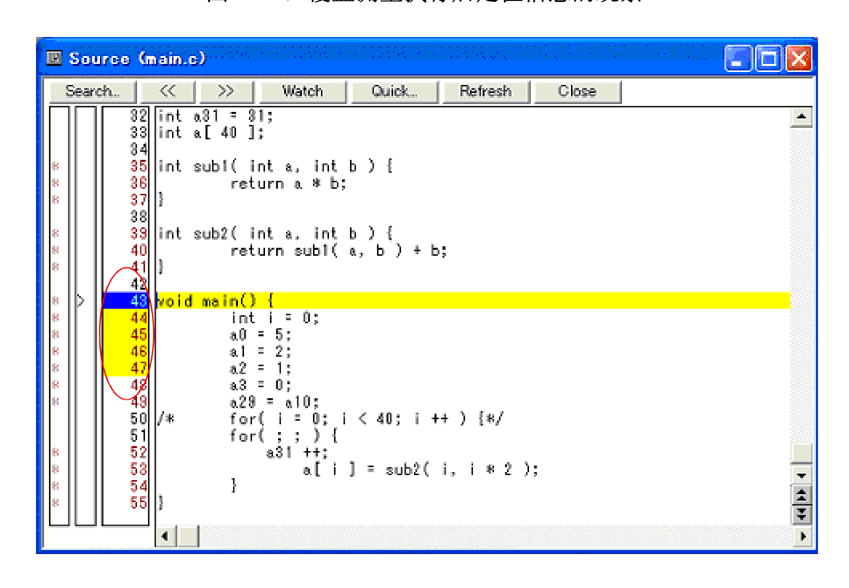
源文件窗口和汇编窗口的显示结果可以存为查看文件(参考"5.15.2窗口显示信息(察看文件)")

已执行覆盖率测量的代码所在地址或行号采用下表所示高亮的方式显示。在查看文件中，下表中的标识将被用来代替背景色被附加到相应的行号或地址上。

表 5-15：覆盖率测量执行部位信息的显示格式

覆盖率	源文件窗口		汇编窗口	
	背景颜色	标记	背景颜色	标记
这一行代码的执行率为100%	黄色	'@'	黄色	'@'
这一行代码执行率为1 to 99%	橙色	'+'		-
这一行代码的执行率为0%	没有高亮显示的颜色	-	没有高亮显示的颜色	-

图 5-21：覆盖测量执行后定位信息的观察



5.11.4 RRM 功能，跟踪功能，覆盖率功能的互斥性

RRM功能，跟踪功能，以及覆盖率测量功能[IECUBE]是互斥的。

因此，当选中RRM功能后，跟踪功能和覆盖率测量功能无法使用。当选中跟踪功能后，RRM功能和覆盖率测量功能无法使用。当选中覆盖率测量功能后，RRM功能和跟踪功能无法使用。

如果需要在三个功能之间切换，在[Opiton]中选择[RRM Function/Trace Function/Coverage Function].其中之一。

一般来说这三个功能是互斥的，然而，下面的几种操作是可能的。

- 功能之间的切换并不清除覆盖率测量数据。
- 即使选中覆盖率测量功能之外的功能，覆盖率测量数据也可以被下载，上传，以及清除。
- 覆盖率测量功能被选中时，如果跟踪模式被锁定在”All PC”时仍可进行跟踪操作。(不允许条件跟踪。)
- RRM被选中后，在跟踪显示窗口中显示的数据仅指示实时监测功能[IECUBE]的采样范围内的跳转信息和访问信息。(在跟踪模式锁定在”Branch PC”的情况下实现跟踪。)

5.12　事件功能

事件指在目标系统的调试中的特定状态，诸如"在地址0x1000处取指"或"在0x2000处写入数据"。

在ID850QB中，这些事件用为每个调试功能的触发器，例如断点和跟踪。

这一节介绍如下功能：

- 使用事件功能
- 创建事件
- 设置事件条件
- 每个事件条件下使能的事件数目
- 管理事件

5.12.1　使用事件功能

在下表中罗列了设置不同的调试方式下事件(事件条件和事件关联条件)包含的条件。作为结果，根据调试目的事件条件可以被使用。

表 5-16 不同的事件条件

事件条件	标记	内容 ->设置对话框
断点事件	B	用户程序的执行或者跟踪停止的条件（参考 "5.4 断点功能"。） ->断点对话框
跟踪事件　**[IECUBE]**	T	用户程序执行过程被存储到跟踪存储区的条件。(参考 "5.10跟踪功能[IECUBE]".) ->跟踪对话框
定时器事件 **[IECUBE]**	Ti	指定测量的起始时间和停止时间的条件(参考"5.9 定时器功能[IECUBE]".) ->定时器对话框

5.12.2　创建事件

通过对事件条件和事件关联条件的登录，事件可以做为前述各种事件条件的执行触发器，独立的命名称为事件。

(1)　创建和登录事件

在事件对话框中创建事件条件。

在此对话框中设置一个地址条件、状态条件和数据条件。这些条件组合在一起成为一个事件条件，命名并且登录此事件条件。

一个简单的方法是同时在源文件窗口和汇编窗口中设置断点作为事件发生的条件(参考"5.4.2 断点设置"。)

(2)　创建和登录事件关联

事件关联条件是指一系列条件的发生具有顺序的限制。用户程序按照指定的顺序发生时产生一个事件关联条件。

在事件关联对话框中可以创建事件关联条件。

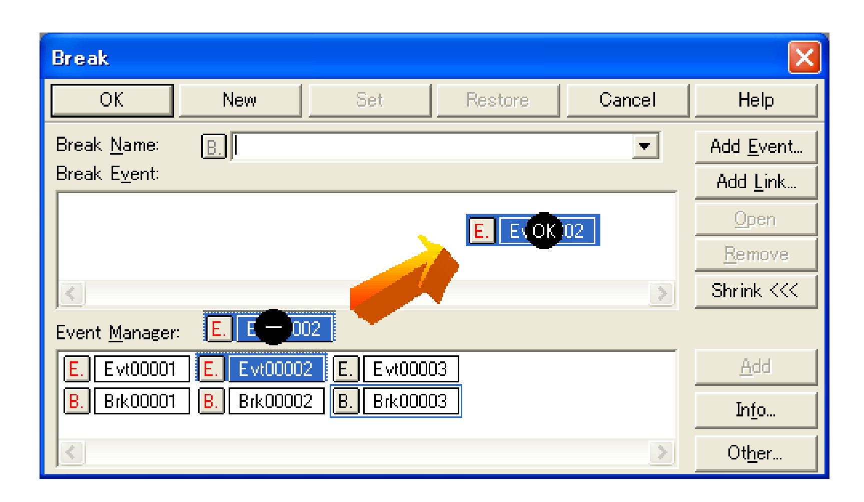
5.12.3　设置事件条件

表5-16中列出的不同的事件条件分别在相应的对话框中创建。

(1)　设置不同的事件条件

不同的事件条件的设置可以通过选中相应设置对话框中的事件管理区域中(或者事件管理器)显示的期望的事件条件或者事件关联条件图标，并且在条件区域中拖放这个图标来实现。

图 5-22：不同事件条件的设置



当鼠标光标拖过可设置的条件区时光标形状变为”OK”。

对于已经创建的事件条件，事件图标变为红色，通过单击设置对话框中<Set>按钮或<OK>按钮进行设置。当事件被设置后，当事件条件发生时调试功能被激活。

(2)　应用选择模式设置(检查内容后设置)

通过把焦点放在被设置的条件区域同时单击<Add Event>按钮或<Add Link...>按钮，事件对话框和事件关联对话框打开为选择模式。因为当在对话框中选择了一个设置条件时，相关的详细条件被显示，在检查这些内容后设置条件。

(3)　复制和移动事件图标

在事件条件设置区域，使用下面的方法，通过拖放操作可以实现事件条件的复制和移动。

- 如果只使用鼠标拖动事件条件，执行事件条件移动操作。
- 如果按下Ctrl键拖动事件条件，执行事件条件复制操作。

(4)　事件管理区的操作

通过把焦点放置到欲设置的事件条件区域，同时选择事件图标，单击<Add>按钮可以设置事件条件。

事件设置内容显示

选择一个事件并且单击<Open>按钮或者双击该事件，打开相应的设置对话框，显示事件的设置内容。

删除

通过选择事件然后单击<Remove/Delete>按钮或删除键可以删除被选择的事件。

改变显示模式和排序

在事件管理区通过选择<Info...>按钮可以改变显示模式和排序。

非显示区域

通过单击<Shrink...>按钮可以隐藏显示区域。

5.12.4　每个事件条件下使能的事件数目

最多可以登录256个作为事件条件或多种事件条件的条件。

事件条件或链事件条件可以被设置为多种类型的事件如断点和跟踪。

然而，可以同时设置(使能)的事件条件有如下限制。

所以，如果超出了有效数目或者使用的事件条件或事件关联条件超出了允许同时使用的最大数目，必须禁止多种事件条件的设置后重新进行登录。(参考"5.12.5事件管理")

表 5-17：每个事件条件下可使能的事件数目

连接的 IE		事件		事件关联	断点	跟踪	定时器
		执行	访问				
[IECUBE]		10[*a]	6[*b]	1	10+6	1	7[*g]
[MINICUBE]	Nx85ET (RCU0+TEU+TRCU)	10[*a]	4[*c]	1[*d]	10+4	-	-
	Nx85E901　(RCU0)，RCU1	2[*e]		1[*f]	2[*e]	-	
[MINICUBE2]	具有调试功能	2[*e]		1[*f]	2[*e]	-	
	没有调试功能	-		-	-	-	

***a**　执行前为 2(仅对断点适用，不可指定地址范围)，执行事件后为 8
(当指定地址范围时为4，2个事件用来指定地址范围)

***b**　当地址范围指定时为3，因为事件2用来指定地址范围

***c**　当地址范围被指定时为2，因为2个事件用来指定地址范围

***d**　第1段至第4段适用

***e**　不能指定地址范围。

***f**　仅能在第1段和第2段中设置

***g**　地址范围被指定给起始事件终止事件，不能设置事件条件。当指定了地址范围后可以设置事件关联条件。

5.12.5　事件管理

在事件管理器中可实现对事件进行管理。

在事件管理器中可以对不同的事件条件进行显示、使能/禁止和删除操作。

图 5-23：事件的管理(事件管理器)

Event Manager

New... | Open | Enable | Delete | Delete All | Info... | Close

E.	Evt00001	[S]EX [A]test.c#23(0x1a52) [M]0x00000000
E.	Evt00002	[S]EX [A]test.c#15(0x1a36) [M]0x00000000
B.	Brk00002	[B]Evt00002
E.	Evt00003	[S]EX [A]test.c#16(0x1a3e) [M]0x00000000
B.	Brk00003	[B]Evt00003
L.	Lnk00001	[P1]Evt00001 [P2]Evt00002 [P3]Evt00003

(1) 事件图标

事件图标包含指示事件类型的名称和标识。事件图标的颜色表示事件的设置状态。

通过单击标识部分实现使能/禁止的切换。

表 5-18：事件图标

字符颜色	标记	含义
红	E.L.	表明用来作为各种不同事件条件的事件条件或者事件关联条件处于激活状态。
	B.T.Ti.	表明各种事件条件是有效的。 各种事件当其条件被满足时发生事件发生。
黑	E.L.	表明用来作各种不同事件条件的事件条件或者事件关联条件处于失效状态。
	B.T.Ti.	表明各种事件条件是无效的。 各种事件即使当其条件被满足时事件也不发生。
黄	E.L.	表明因为不能被目前调用的程序识别，为某个事件指定的符号被保留。
	B.T.Ti.	表明各种事件条件被保留。 各种事件即使条件满足也不会发生。

5.13　RRM 功能

这一节介绍下列与RRM功能有关项。

- 实时监视功能[IECUBE]
- 伪实时监视功能(Break When Readout)

注意事项:　RRM功能，跟踪功能[IECUBE]和覆盖率测量功能[IECUBE]是互斥的。(参考"5.11.4 RRM功能，跟踪功能，覆盖率测试功能是互斥的")

在[Option]菜单中实现三种功能之间的切换。

5.13.1　实时监视功能[IECUBE]

表5-19列出实时监视功能中可以被加载的数据范围。

在观察窗口和存储器窗口中可以一直实时显示分配到这个区间的变量和数据。

在扩展选项对话框中可以设置采样间隔。

表 5-19：实时监视功能中可以被采样的区域

连接的 IE	采样范围
[IECUBE]	在RRM 设置对话框中指定的区域

图 5-24：RRM 设置对话框

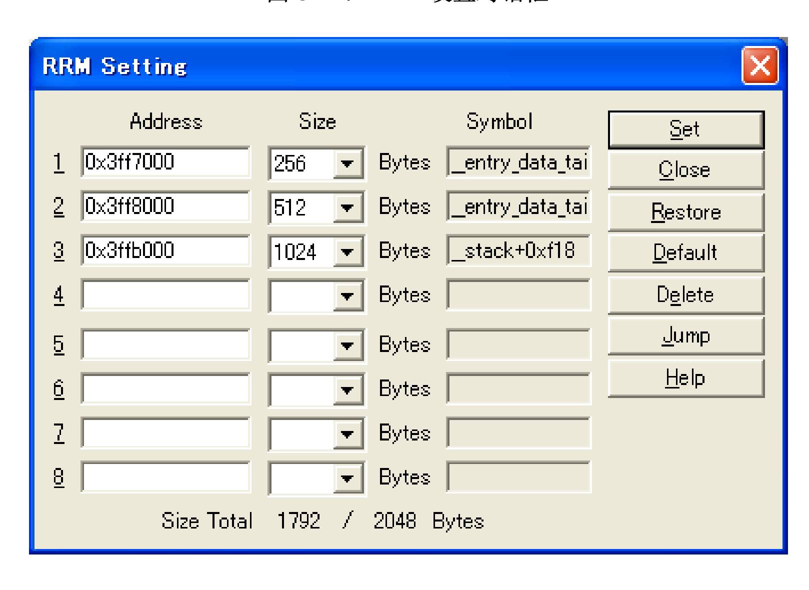
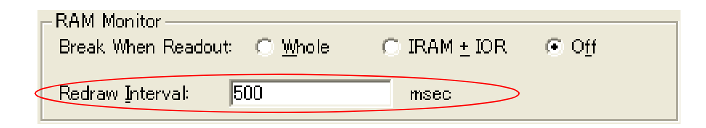


图 5-25：使用实时监视功能指定采样间隔



5.13.2　伪实时监视功能(读出时产生断点)

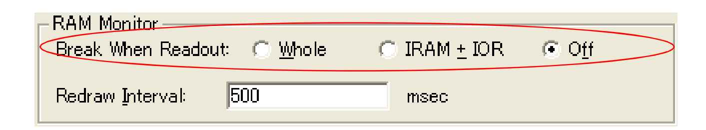
对于实时监控功能不能读取的区域，采用伪实时监视功能来读取。

当执行伪实时监视功能时，存储器内容被软件仿真器读取。因此在读取时用户程序会产生短暂的中断。

分配到此区域的变量和数据可以接近实时地显示在观察窗口和存储窗口中。

在扩展选项对话框中可以打开/关闭伪实时监视功能和指定采样范围。

图 5-26：伪实时监视功能的设定



注意事项:　伪实时监视功能和软件中断功能不能同时执行。当伪实时监视功能使能时("Off"之外的项目被选中)，不能设置软件断点。即使已经设置过的软件断点功能也被屏蔽。

5.14　DMM 功能

DMM(动态存储器修改)是在用户程序执行期间实时修改存储器(RAM)的内容。

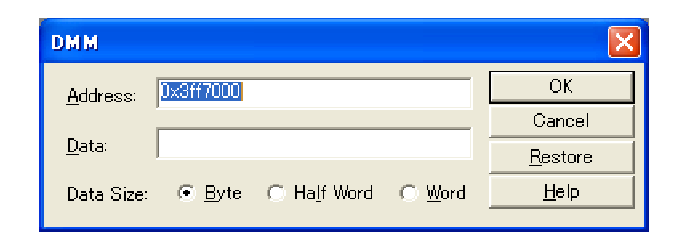
在存储器窗口、寄存器窗口或IOR窗口单击<DMM…>可以打开DMM对话框。设置DMM目标地址和数据。

注意事项**1:**　因为这个功能是通过软件仿真来实现的，它利用用户程序执行的停止瞬间实现存储器的写入(伪DMM功能)。

注意事项**2:**　DMM功能和其他的软件断点功能不能同时使用。

当一个合法的软件断点被设置，在用户程序执行期间写入(DMM)是被禁止的(如果执行将出现错误的结果)。

图 5-27：修改存储器内容(DMM 对话框)



5.15　加载/存储功能

ID850QB允许以文件的方式加载和存储以下类型的信息。因此，这些信息是可以恢复的。

- 调试环境(工程文件)
- 窗口显示信息(工程文件)
- 窗口设置信息(设置文件)

备注: 简单的窗口状态可以通过选择[Window]菜单下[Static]功能保持(参考"5.16.1 激活状态和静止状态")。

5.15.1　调试环境(工程文件)

工程文件(*.prj)是记录调试环境信息的文件。
当调试环境保存时会产生一个工程文件，在随后的应用中调试环境可以通过加载工程文件来恢复。
工程文件的加载和保存是在工程文件保存对话框和工程文件加载对话框中分别进行的。启动时要加载工程文件通过单击配置对话框中<Project>按钮实现。
下表内容是工程文件中保存的信息：

表 5-20：保存在工程文件中的内容

窗口名称	保存的内容
设置对话窗口	所有项目(目标设备，时钟设置，引脚屏蔽(pin mask)设置，映射信息)
主窗口	显示位置，工具条/状态条/按钮显示信息，运行模式信息
下载对话窗口	下载文件信息
扩展选项对话窗口 调试选项对话框 故障保险断点对话框 闪存选项对话框 数据闪存选项对话框 RRM 设置对话框	设置信息
汇编窗口 存储器窗口	窗口显示信息，显示开始地址
源文件窗口 堆栈窗口 IOR(IO寄存器)窗口 局部变量窗口 跟踪显示窗口 代码覆盖率窗口 事件管理器 控制台窗口 扩展窗口 寄存器窗口	窗口显示信息
事件对话框 事件关联对话框 断点对话框 跟踪对话框 定时器对话框	窗口显示信息，事件信息
列表窗口	窗口显示位置
观察窗口	窗口显示信息[注]，观察登录信息
添加 I/O 端口对话框	添加的 I/O 口信息
DMM 对话框	DMM 信息
延时计数对话框	延时计数值
软件断点管理器	窗口显示信息，软件断点信息

注:　　结构指针成员、数组指针成员的显示状态，每个成员的显示基点等等，这些信息是不保存的。

5.15.2 窗口显示信息(查看文件)

查看文件用来记录窗口显示信息。

每个窗口都可以存储和加载查看文件。

当一个查看文件加载时，将显示一个参考窗口(静态的源文件窗口)，同时将显示保存时刻的信息。

通过查看文件加载对话框和查看文件存储对话框分别实现观察文件的加载和存储。

表 5-21：观察文件的类型

文件类型	当前窗口名字，文件名
源文件 (*.svw)	源文件窗口**注1**
Assemble汇编文件 (*.dis)	汇编窗口**注1**
Memory存储器文件 (*.mem)	存储器窗口
Watch观察文件 (*.wch)	观察窗口
Register 寄存器文件 (*.rgw)	寄存器窗口
I/O RegisterI/O寄存器文件 (*.ior)	IOR窗口
Local Variable局部变量文件 (*.loc)	局部变量窗口
Stack Trace堆栈跟踪文件 (*.stk)	堆栈窗口
Trace 跟踪文件(*.tvw)	跟踪显示窗口
Code Coverage代码覆盖率文件 (*.csv)	代码覆盖率窗口 (tab选中的数据被分别保存)
List列表文件 (*.csv)	列表窗口 (Tab键选中的数据被单独存放)
Console 控制台文件 (*.log)	控制台窗口
All 所有类型文件(*.*)	所有文件
Source源文件 (*.c, *.s)	源文件**注2**
Text 文本文件(*.txt)	文本文件

注1:　　指示代码覆盖率测量结果(执行/没有执行)的标识增加到显示文件的内容中(参考"表5-15执行代码覆盖率测量的观察定位格式")**[IECUBE]**。

注2:　　在扩展选项对话框中可以改变源文件的扩展名称。

5.15.3　窗口设置信息(设置文件)

设置文件是用来保存窗口设置信息(观察数据设置、周边I/O寄存器设置和事件设置)的文件。

对于每个窗口可以加载和保存设置文件。

当加载一个设置文件时，将显示目标窗口，同时被保存的设置信息被恢复。

在环境设置文件加载对话框和环境设置文件保存对话框中分别实现对设置文件的加载和保存。

表 5-22：设置文件的类型

文件类型	当前窗口名字
Watch 观察文件(*.wch)Note	观察窗口
I/O RegisterI/O寄存器文件(*.ior)Note	IOR　窗口
Event 事件文件(*.evn)	事件管理器

注:　　无法加载一个变量值。

5.16　窗口的公共功能

每个窗口具有下列公共功能：

- 每个窗口的公共功能
- 跳转功能
- 关联窗口的跟踪结果[IECUBE]
- 拖&放功能
- 注意事项

5.16.1 激活状态和静止状态

下面所列的窗口都有两个状态：激活状态和静止状态

- 源文件窗口(读入符号信息显示源文件的窗口)
- 汇编窗口
- 存储器窗口

在激活状态只能打开一个窗口。但是，因为可以打开两个或多个静态窗口，因此窗口的当前状态可以被临时保持。

通过[Window]菜单选择窗口状态。

(1) 激活状态

激活窗口中显示位置和显示内容随着当前PC的值自动更新。

在跳转功能中可以跳转到目的窗口。如果这个窗口和跟踪观察窗口相关联，激活窗口中显示的内容随着关联的跟踪观察窗口自动修改。

在激活状态下只有一个窗口可以被打开。

(2) 静止状态

静止窗口中显示位置不能随着当前 PC 的值移动，但是更新显示内容。

静止状态窗口不能作为跳转功能中的目的，而且不和跟踪观察窗口相关联。

如果一个激活窗口已经打开，下一个窗口以静止状态打开。

在同一时刻可以打开两个或更多静止窗口。

5.16.2　跳转功能

这个功能可以实现从光标所指的行和地址(跳转指针)跳转到下列窗口。跳转到的窗口将显示在跳转指针。

- 源文件窗口
- 汇编窗口
- 存储器窗口

用户可以在上述窗口间实现跳转，或者从跟踪观察窗口、堆栈窗口、事件管理器和寄存器窗口跳到上述窗口中。

(1)　跳转方式

跳转方式描述如下：

1)　移动光标到用作跳转指针的行或地址处，在可以实现跳转的窗口上(选择事件管理器中的事件图标)。

2)　从[Jump]菜单选择随后的菜单项执行跳转。

注意事项:　如果光标所在行没有程序代码，上面或下面紧邻行将被用作跳转指针。

(2)　跳转源地址细节

跳转源地址细节描述如下：

表 5-23：跳转源地址细节

目标窗口	跳转指针的细节	
从寄存器窗口	寄存器被选择	
从存储器窗口	光标处的地址	
从事件管理器	如果选中的事件图标是一个事件条件，一个地址条件被用作跳转指针	
	如果某点设置了地址条件	跳至指定的地址
	如果地址条件设为某范围	跳转至低端地址(如果指定了屏蔽,则在其前面的点地址)
	如果地址条件设置为某位(bit)	跳转至该位地址

目标窗口	跳转指针的细节	
从堆栈窗口	光标所在位置的堆栈结构编号所指示的函数将被作为跳转指针使用。	
	在当前函数的情况下	
	如果目的是源文件窗口	跳至当前的 PC 行
	其他情况	跳至当前的 PC 地址
	在其他函数的情况下	
	如果目的地址是源文件窗口	跳至调用嵌套函数的所在行
	其他情况	跳至调用嵌套函数的指令的相邻地址
从跟踪显示窗口	跳至存储器窗口	
	如果光标位置在一个访问地址，访问数据，或者访问状态	访问地址
	其他情况	取指地址
跳至源文件窗口 或 汇编窗口	取指地址	

5.16.3 关联窗口的跟踪结果[IECUBE]

通过关联跟踪观察窗口和其他窗口(源文件窗口、汇编窗口或存储器窗口)相应的内容在关联的窗口显示，这种显示通过使用跟踪观察窗口中的光标地址指针来实现。

如果光标移动到跟踪显示窗口，关联窗口中相应的部分高亮显示或者由光标位置指示。

(1) 关联方法

关联方法如下：

1) 把跟踪观察窗口设置为当前窗口。
2) 在[View]菜单中[Window Synchronize]选择被关联的窗口。
3) 移动光标到跟踪观察窗口的跟踪结果显示区域的某一行实现关联。
4) 使用3)中选择的行地址作为指针，相应的内容在2)中选择的窗口的显示区域以高亮的方式显示(或者由光标位置指示)。

备注: 如果存储器窗口被关联，关联的源文件窗口和下面依赖光标位置的跟踪结果显示区域的内容是不同的：

如下所示，如果存储窗口被关联，连接的源地址由跟踪结果显示区域中的光标位置确定。

-访问地址，访问数据，访问状态->访问地址

-其他->取指地址

当源文件窗口或汇编窗口被关联时，取指地址总是被用作指针。

5.16.4 拖&放功能

可以采用下面的方法把选择并高亮显示的行号、地址和文本拖放功能至另一个窗口。

1) 拖动选择的行号、地址和文本。

->这时鼠标的形状由箭头变为”-”。

2) 把在窗口中选中的内容放至可以释放的窗口或区域。

->当鼠标移过一个可以放下选中内容的窗口或选择区域时，鼠标形状由”-”变为”OK”。

在拖放了一个地址行号的窗口中，对放置的地址或者从放置的行号获得的地址产生操作。例如，位于源文件窗口中的变量可以简单地通过拖放到观察窗口中来登记。

(1) 拖放的细节

根据地址和行号所放置的窗口或区域的不同，拖动和放置后执行的操作是不同的。

表 5-24：拖动和放置功能的细节(地址/行)

放置的目的窗口/区域	放置后的操作
每个多种事件设置对话框中的事件管理器和事件管理器区域	通过使用放下位置行号或者地址号作为一个地址条件自动地创建一个执行事件条件。事件条件名字自动地创建为Evt00001，Evt00002，等等。地址条件的符号设置接近于符号名称+偏移值。没有指定执行计数。
每个多种事件设置对话框(地址和数据设置区域除外)中的条件设置区域	通过使用放下位置行号或者地址号作为一个地址条件自动地创建一个执行事件条件。自动创建的执行事件条件的设置在每个放置行号或者地址处的条件设置区域进行。事件条件名字自动地创建为Evt00001，Evt00002，等等。地址条件的符号设置接近于符号名称+偏移值。没有指定path count。
每个多种事件设置对话框(地址和数据设置区域)中的条件设置区域	在每个放置行号或者地址区域设置行号或者地址的文本。地址条件的符号设置接近于符号名称+偏移值。

表 5-25：拖动和放置细节(字符串特征)

<table>
<tr><th>放置的目的窗口/区域</th><th>放置后的操作</th></tr>
<tr><td>每个多种事件设置对话框中的事件管理器和事件管理器区域</td><td>如果放下的文本可以从符号转换为地址值，一个访问(R/W)状态(所有访问状态)或者执行状态的事件条件被创建，此事件使用转换的地址值作为一个地址条件。地址条件的符号设置接近于符号名称+偏移值。没有指定path count。
地址条件由放置的文本设置。
要创建的事件条件和符号之间的关系如下：
<table>
<tr><th>Symbols符号</th><th>Status状态</th></tr>
<tr><td>Variable变量</td><td>Access访问(R/W)</td></tr>
<tr><td>Function功能</td><td>Execute执行</td></tr>
<tr><td>Symbol in data section数据区域的符号</td><td>Access访问(R/W)</td></tr>
<tr><td>Symbol in code section代码区域的符号</td><td>Execute执行</td></tr>
<tr><td>Others其它</td><td>Access访问(R/W)</td></tr>
</table></td></tr>
<tr><td>每个多种事件设置对话框(地址和数据设置区域除外)中的条件设置区域</td><td>如果放下的文本可以从符号转换为地址值，一个访问(R/W)状态(所有访问状态)或者执行状态的事件条件被创建，此事件使用转换的地址值作为一个地址条件。自动创建的执行事件条件的设置在每个放置行号或者地址处的条件设置区域进行。此事件使用转换的地址值作为一个地址条件。事件条件名字自动地创建为Evt00001，Evt00002，等等。
地址条件由放下的文本设置。
要创建的事件条件和符号之间的关系如下：
<table>
<tr><th>Symbols符号</th><th>Status状态</th></tr>
<tr><td>Variable变量</td><td>Access访问 (R/W)</td></tr>
<tr><td>Function功能</td><td>Execute执行</td></tr>
<tr><td>Symbol in data section数据区域的符号</td><td>Access访问 (R/W)</td></tr>
<tr><td>Symbol in code section代码区域的符号</td><td>Execute执行</td></tr>
<tr><td>Others其它</td><td>Access 访问(R/W)</td></tr>
</table></td></tr>
<tr><td>每个多种事件设置对话框(地址和数据设置区域)中的条件设置区域</td><td>放置的文本在此区域设置。</td></tr>
<tr><td>Watch Window观察窗口</td><td>放置的文本被识别为一个符号，符号的内容被显示。</td></tr>
</table>

备注:　　下面是各种事件设置窗口。

-事件对话框

-事件关联对话框

-断点对话框

-跟踪对话框

-定时器对话框

5.16.5　注意事项

(1)　在每个窗口的一行中可以显示的字符数为319。

(2)　若显示区域过窄，将不能正常显示。在这种情况下可以增加显示窗口的宽度。

(3)　在一个激活窗口中当光标位置被改变后，对窗口执行<Refresh>操作可能不会成功。单击<Refresh>按钮重新绘制窗口。

(4)　当光标位于某个窗口时，按下”F1”键可以打开该窗口的帮助文件。但是当没有跟踪结果可以显示时，光标不能放在跟踪观察窗口上，例如在刚刚启动时，即使按下”F1”键也不可能出现帮助窗口。在这种情况下可以从[Help]菜单选择[Current Window Help]来实现帮助。

(5)　在Go &Go执行期间不要从[Run]菜单选择[Slowmotion]操作，因为在 Go & Go 操作期间[Run]菜单中的[Slowmotion]通常无效，但是有一个时刻可以选择它，因此在这个时刻如果选择”Slowmotion”，即使选择“Run”菜单中的”STOP”功能(或点击STOP按钮)也不能使执行的程序停止。

(6)　如果因为某种原因在事件图标正在拖动期间应用程序发生切换，事件图标不允许再被放置。利用”ESC”键可以脱离拖动操作，然后重新尝试拖动操作。

第 6 章　窗口参考

这一章详细介绍ID850QB窗口和对话框中的功能.

6.1　窗口列表

ID850QB的窗口列表如下。

表 6-1 窗口列表

窗口名称	内容
主窗口	当ID850QB启动时，这个窗口首先显示。它控制用户程序的执行。可以从这个窗口打开其他窗口。
配置对话框	显示和设置ID850QB操作环境
扩展选项对话框	显示和设置ID850QB扩展选项
故障保险断点对话框	设置故障保险断点
RRM设置对话框	设置RRM采样范围
闪存选项对话框	设置闪存自编程仿真 **[IECUBE]**
数据闪存选项对话框	数据闪存错误仿真设置 **[IECUBE]**
调试器选项对话框	显示和设置其他操作
工程文件保存对话框	保存当前的调试环境至工程文件
工程文件加载对话框	加载调试环境
下载对话框	下载
上载对话框	上载
加载模块列表对话框	下载的加载模块文件名称列表
源文件窗口	显示源文件和文本文件
源文件搜索对话框	源文件窗口中搜索
源文件文本移动对话框	指定的在源文件窗口中显示文件和指定文件的显示起始地址
汇编窗口	程序反汇编和执行在线汇编
汇编搜索对话框	汇编窗口中搜索
地址移动对话框	指定显示存储器窗口、汇编窗口和IOR窗口内容的起始地址
符号地址转换对话框	显示一个指定变量、函数的地址或符号的数值
列表窗口	函数、变量、符号、段和中断请求列表
观察窗口	显示和修改指定的观察数据
快速观察对话框	显示临时指定的观察数据
添加观察项对话框	在观察窗口登录观察数据
改变观察对话框	修改显示在观察窗口中的观察数据
局部变量对话框	在当前函数中显示和修改局部变量

窗口名称	内容
堆栈窗口	显示当前堆栈的内容
存储器窗口	显示存储器的内容
存储器搜索对话框	存储器窗口中的搜索
存储器填充对话框	用指定内容填充存储器
存储器复制对话框	存储器复制
存储器比较对话框	存储器比较
存储器比较结果对话框	显示存储器比较的结果
DMM对话框	设置DMM的地址和数据
寄存器登录窗口	显示寄存器的内容
寄存器登录选择对话框	在寄存器窗口要显示的寄存器
IOR窗口	显示IOR的内容
IOR选择对话框	IOR窗口中选择要显示的周边I/O寄存器和I/O端口
增加I/O端口对话框	登录在IOR窗口显示的I/O端口
定时器对话框	登录和设置定时器事件条件，显示执行时间的测量结果 **[IECUBE]**
定时器结果对话框	显示执行时间测量结果　**[IECUBE]**
跟踪观察窗口	显示跟踪结果**[IECUBE]**
跟踪搜索对话框	跟踪数据搜索**[IECUBE]**
跟踪数据选择对话框	在跟踪观察窗口选择需要显示的项目 **[IECUBE]**
跟踪移动对话框	指定显示跟踪观察窗口内容的起始地址 **[IECUBE]**
跟踪对话框	跟踪事件条件登录和设置**[IECUBE]**
延迟计数对话框	设置延时触发跟踪事件的延时计数**[IECUBE]**
代码覆盖率窗口	显示代码覆盖率结果**[IECUBE]**
覆盖地址对话框	选择代码覆盖测量范围 **[IECUBE]**
软件断点管理器	显示，使能，禁止，和删除软件断点
事件管理器	显示，使能，禁止，和删除事件条件
事件对话框	登录事件条件
事件关联对话框	登录事件关联条件
断点对话框	登录和设置断点事件条件
观察文件保存对话框	保存当前窗口的显示信息为查看文件。
查看文件加载对话框	对每个窗口加载查看文件。
环境设置文件保存对话框	保存当前窗口的设置信息到设置文件.

窗口名称	内容
环境设置文件加载对话框	为每个窗口加载窗口设置文件.
复位调试器对话框	初始化ID850QB、CPU和符号信息
退出调试器对话框	停止ID850QB.
关于对话框	显示ID850QB版本
控制台窗口	输入命令
浏览对话框	选择被设置的文件

6.2　窗口解释

这一节解释下列每个窗口或对话框:

窗口名称/ 对话框名称

简单介绍窗口和对话框的功能和须注意的要点。

如果一个和IE关联的无效对话框或窗口已经存在，有效连接的IE的名字会在窗口/对话框的名称的右下方显示。

另外，窗口和对话框中显示的图形也有图解。

相关的操作项目也有解释。

打开

解释如何打开窗口和对话框。

每个区域解释

解释窗口和对话框每个区域设置和显示的内容

上下文菜单

解释当鼠标单击右键时在窗口中显示的上下文菜单，在上下文菜单中，窗口中经常用到的功能可以很方便的被选中(只有窗口方式)。

相关的操作

解释和这个窗口和对话框关联的窗口和对话框的操作方式。

主窗口

当ID850QB启动并初始化时，这个窗口是自动打开的。

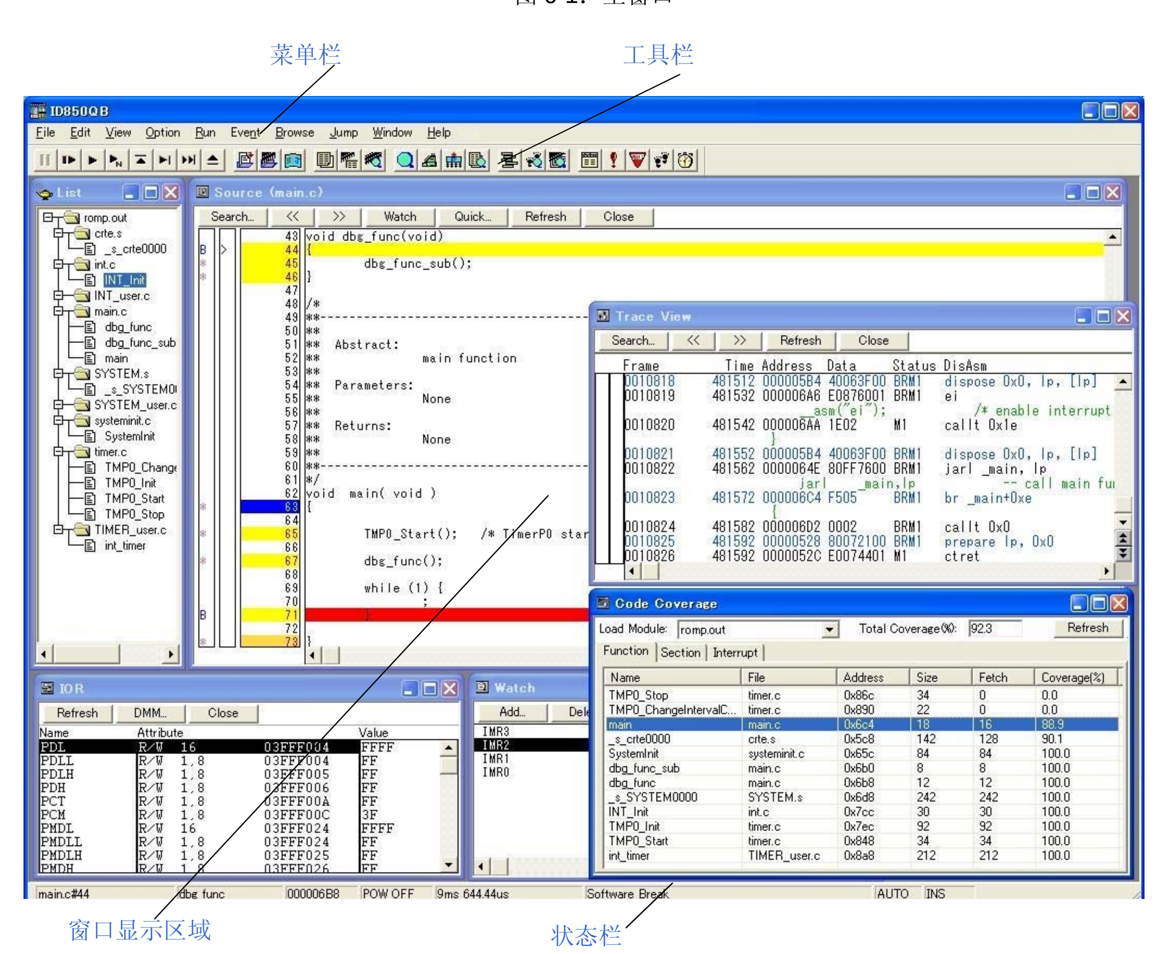
在ID850QB中，在主窗口中操作其它窗口 (参考"6.1 窗口列表")。

在这个窗口中执行用户程序。

用户程序的执行可以通过以下三种方式控制:

- 源程序模式 (在源程序级调试用户程序)
- 指令模式 (在指令级调试用户程序)
- 自动模式 (自动选择源程序模式和指令模式) (缺省方式)

图 6-1：主窗口



菜单栏

(1) [File] 文件菜单
(2) [Edit] 编辑菜单
(3) [View] 显示菜单
(4) [Option] 选项菜单
(5) [Run] 运行菜单
(6) [Event] 事件菜单
(7) [Browse] 浏览菜单
(8) [Jump] 跳转菜单
(9) [Window] 窗口菜单
(10) [Help] 帮助菜单

(1)　[File] 菜单

Open... 打开	打开查看文件、源文件和文本文件。打开查看文件加载对话框。操作因对话框中选择的文件扩展名不同而不同。
Save As... 存为	把当前窗口显示的内容保存到一个指定的文件中。 打开查看文件保存对话框。
Close关闭	关闭当前窗口
Download... 下载	下载一个文件。打开下载对话框
Load Module...加载模块	对已经下载的文件名称进行列表。 打开加载文件列表对话框。
Upload... 上传	上载程序，打开上载对话框。
Project工程	操作一个工程文件
Open... 打开	打开一个工程文件 打开工程文件加载对话框
Save保存	ID850QB读入的工程文件被当前的状态覆盖.
Save As... 存为	保存当前的状态到指定的工程文件。 打开工程文件保存对话框。
Environment环境	操作一个设置文件
Open... 打开	打开一个设置文件 打开环境设置文件加载对话框
Save As... 存为	保存当前窗口设置的内容到设置文件。 打开环境设置文件保存对话框。
Debugger Reset... 调试器复位	初始化CPU，符号，和ID850QB。 打开复位调试器对话框。
Exit退出	终止ID850QB. (参考"3.4 终止".) 打开退出调试对话框
(Open file) 打开文件	打开文件名称列表

(2) [Edit] 菜单

Cut剪切	剪切选择的字符串并存入剪贴板的缓冲区中。
Copy复制	复制选择的字符串并存入剪贴板的缓冲区中。
Paste粘贴	将剪贴板的缓冲区的内容粘贴到光标所在的位置。
Write in写入	把修改内容写入目标
Restore恢复	取消修订
Memory存储器	存储器内容操作
Fill填充...	用指定的代码填充存储器。打开存储器填充对话框。
Copy复制...	复制存储器，打开存储器复制对话框。
Compare比较...	存储器比较，打开存储器比较对话框。
DMM...	在用户程序的实时执行期间重写存储器的内容。打开DMM对话框。
Edit Source源文件编辑	PM+运行时，使用PM+指定的编辑器打开激活的源文件窗口中显示的源文件。

(3) [View] 菜单

[View]菜单包含按照每个激活窗口(设置)加入的通用部分以及专用部分。有关专用部分的详细内容，参考每个窗口的描述。

(a) 公共项

Search搜索...	执行搜索 打开当前窗口项对应的搜索对话框。 单击<Search>按钮执行同样的操作。
Move移动...	移动显示位置。 打开当前窗口相应的指定对话框。
Quick Watch快速观察...	临时显示指定数据的内容。 打开快速观察对话框。
Add Watch添加观察...	登录指定的数据到观察窗口。 打开添加观察对话框。
View Watch显示观察	添加选择的数据到观察窗。 如果数据是一个符号，根据调试选项对话框的设置进行添加。
Change Watch改变观察...	改变观察窗口中选择行中的数据。 打开改变观察对话框。 这个菜单只有在观察窗口中选中一个变量时才是有效的。
Delete Watch删除观察	删除观察窗口中选择的观察点。 这个菜单只有在观察窗口中选择一个变量才是有效的。
Symbol符号...	显示指定变量或符号的地址，或者显示指定符号的值。 打开符号地址转换对话框。

(4) [Option] 菜单

工具栏	工具栏显示切换开关
状态栏	状态栏显示切换开关
按钮	窗口按钮显示切换开关
源程序模式	源程序级别的单步执行 (以行为单位).
指令模式	指令级上的单步执行(以指令为单位).
自动模式	自动选择源程序级的单步执行或者指令级的单步执行(缺省)。 如果源文件窗口是激活窗口，在源程序级单步执行(除了混合显示模式以外) 如果汇编窗口是激活窗口，执行指令级的单步执行操作。 如果没有任何一个窗口是激活的，单步执行在源程序级进行。
Configuration设置...	设置环境，打开配置对话框。
Extended Option扩展功能...	设置扩展选项，打开扩展选项对话框。
RRM 设置... **[IECUBE]**	设置RRM功能的采样范围。 打开RRM设置对话框
Flash Option闪存选项... **[IECUBE]**	这个对话框用来实现闪存自编程仿真功能的设置。 打开闪存选项对话框。
Data Flash Option数据闪存选项... **[IECUBE]**	这个对话框用来实现数据闪存错误仿真功能的设置。 打开数据闪存选项对话框。
Debugger Option调试器选项...	设置ID850QB选项。打开调试器选项对话框
Add I/O Port添加I/O 端口...	添加用户定义I/O端口。打开增加I/O端口对话框
Trace Clear跟踪清除 **[IECUBE]**	清除跟踪数据。 这一项只有当跟踪观察窗口处于激活状态下才能显示。
Coverage覆盖率**[IECUBE]**	打开覆盖率测量相关的对话框。
Clear清除**[IECUBE]**	清除覆盖测量结果。
Select选择... **[IECUBE]**	选择1MB或者更多的覆盖测量范围。 打开覆盖地址对话框
RRM 功能**[IECUBE]**	选择RRM功能 (参考"5.13 RRM 功能"。)(缺省) 这个功能和跟踪功能和覆盖率功能是互斥的(参考"5.11.4 RRM功能，跟踪功能，覆盖率功能的互斥性") 当这些内容被选择时，下列菜单不能被选择。 [Trace start/Trace end], [Uncond. Trace ON/Cond.Trace ON], [Tracer Control Mode]
Trace 功能**[IECUBE]**	选择跟踪功能。 (参考"5.10 跟踪功能[IECUBE]") 这个功能和跟踪功能和覆盖功能是互斥的(参考"5.11.4 RRM功能，跟踪功能，覆盖功能的互斥性") 当这个项目被选择后，下面的菜单不能被选择 [RRM Setting...]

Coverage function 覆盖率功能 [IECUBE]	选择覆盖率功能(参考"5.11 覆盖率测量功能[IECUBE]".) 这个功能和跟踪功能和覆盖功能是互斥的(参考"5.11.4 RRM功能，跟踪功能，覆盖率功能的互斥性") 没有覆盖板时，这些功能是禁止的。 当选择这个功能时，不能选择下面的菜单 [Trace start/Trace end], [Uncond. Trace ON/Cond.Trace ON], [Tracer Control Mode], [RRM Setting...]

(5) [Run] 菜单

Restart复位	复位CPU和执行程序 这个按钮执行同样的操作
Stop停止	强制停止程序执行 这个按钮执行同样的操作.
Go运行	从当前的PC处开始执行程序 这个按钮执行同样的操作.
Ignore break points and Go 忽略设置的断点并执行程序	忽略设置的断点并执行程序 (无论硬件还是软件) 这个按钮执行同样的操作
Return Out返回	执行用户程序直到返回 这个按钮执行同样的操作 .注: 这个命令用于C语言描述的函数中。
Step In单步执行	单步执行程序指令 (单步执行)。 如果调用一个函数或者子程序，其指令也是单步执行。 这个按钮执行同样的操作。
Next Over步越	单步执行程序指令(后续单步执行). 如果调用一个函数或者子程序，不单步执行其指令 这个按钮执行同样的操作.
Start From Here从此处开始	在源文件窗口或汇编窗口中从光标处执行程序。
Come Here运行到	在源文件窗口或汇编窗口中从当前PC处执行程序到光标处。
Go & Go不间断运行	继续执行程序. 如果一个断点条件满足，发生断点事件，窗口刷新并且再次执行程序。 每次单击按钮执行同样的操作。
Slowmotion	继续单步执行程序。 每次单步执行程序后，窗口刷新并且再次执行程序 每次断点事件发生后，点击这个健进新同样的操作。
CPU Reset复位	复位CPU. 这个按钮执行同样的操作

Change PC改变PC指针	将源文件窗口或者汇编窗口中光标所在位置地址设为PC值。
Break Point断点	在源文件窗口或者汇编窗口中光标所在位置处设置或者删除断点。
Software Break Point软断点	在源文件窗口或者汇编窗口中光标所在位置处设置或者删除软断点。
Delete All Breakpoints删除所有断点	删除所有设置的断点事件。
Uncond. Trace ON **[IECUBE]**	使无条件跟踪生效以便在程序执行期间跟踪一直进行。(缺省) 此时，设置的跟踪事件条件被忽略。 当跟踪器激活时，不能被更改跟踪模式。
Cond. Trace ON **[IECUBE]**	在程序执行期间使跟踪事件条件下的条件跟踪和其他跟踪有效。 当跟踪器激活时，不能被更改跟踪模式。
Tracer Control Mode跟踪控制模式 **[IECUBE]**	设置跟踪控制模式。

Non Stop不停止	循环跟踪跟踪内存并覆盖最旧的页面(默认情况)。
Full Stop完全停止	跟踪所有的跟踪内存然后停止跟踪。
Full Break完全断点	跟踪所有的跟踪内存然后停止跟踪与程序执行
Delay Trigger Stop延时触发停止	根据延时计数帧的数目跟踪数据，当延时触发事件发生时停止跟踪。

Timer Start/Timer Stop **[IECUBE]定时器开始/停止**	定时器停止时启动定时器，或者运行时停止。当程序没有执行或者定时器事件没有使用时，此选项无效。当程序开始执行后，定时器测量立即开始进行。
Tracer Start/Tracer Stop跟踪器开始/停止 **[IECUBE]**	跟踪器停止后，启动跟踪器，或者运行时停止，程序停止时此选项无效。当程序开始执行后，跟踪器立即开始进行。

(6)　[Event] 菜单

Event Manager事件管理器	管理不同的事件条件，打开事件管理器。 此按钮执行相同操作。
Software Break Manager 软断点管理器	管理软断点事件条件，打开软断点管理器
Event...事件	登录一个事件条件，打开事件对话框 此按钮执行相同操作
Event Link...事件关联	登录一个事件关联条件。打开事件关联对话框。
Break...断点	登录并设置一个断点事件条件。打开断点对话框 此按钮执行相同操作

C
H

Trace...跟踪 **[IECUBE]**	登录并设置一个跟踪事件条件。打开跟踪对话框。 此按钮执行相同操作
Timer...定时器 **[IECUBE]**	登录并设置一个定时器事件条件。打开定时器对话框。 此按钮执行相同操作
Delay Count... **[IECUBE]**	设置延时计数。 打开延时计数对话框。

(7) [Browse] 菜单

List列表	列表功能，变量，符号，章节，中断要求。 打开列表窗口
Source Text源文本	显示一个源文本，打开源文件窗口 如果此窗口已打开为激活状态，将以静态打开。 此按钮执行相同操作。
Assemble汇编	显示反汇编结果。打开汇编窗口 如果此窗口已打开为激活状态，将以静态打开。 此按钮执行相同操作。.
Memory存储器	显示存储器内容。打开存储器窗口。如果此窗口已打开为激活状态，将以静态打开。 此按钮执行相同操作。
Watch观察	显示观察内容。打开观察窗口 此按钮执行相同操作。
Register寄存器	显示寄存器内容。打开寄存器窗口。此按钮执行相同操作。
I/O Register寄存器	显示周边I/O寄存器内容。打开IOR窗口。此按钮执行相同操作。
Local Variable局部变量	显示局部变量。打开局部变量窗口。.此按钮执行相同操作。
Stack Trace堆栈跟踪	显示堆栈跟踪结果。打开堆栈窗口。 此按钮执行相同操作。
Trace 跟踪 **[IECUBE]**	显示跟踪结果。打开跟踪观察窗口。 此按钮执行相同操作。
Code Coverage 代码覆盖率 **[IECUBE]**	显示代码覆盖率测量结果。打开代码覆盖率窗口 此按钮执行相同操作
Console控制台	打开控制台窗口
Others其它	显示其它窗口(参考"附录A扩展窗口"。)显示一个用户-定义窗口列表

(8)　[Jump] 菜单

Source Text源文本	显示相应的源文本和源代码行，使用当前窗口中选中的数据值作为跳转的目的地址。 然而，如果在跳转目的地址没有行信息存在，无法跳转。 打开源文件窗口。 如果激活的源文件窗口为打开状态，此窗口在最前面显示(所以可以操作)
Assemble汇编	反汇编并且显示以在当前窗口中选中的数据值指定的跳转目的地址所指结果。 打开汇编窗口。 如果汇编窗口以激活状态打开，此窗口在最前面显示(所以可以被操作)
Memory存储器	显示当前窗口选中的数据值所指定的跳转目的地址所指的存储器内容。

(9)　[Window] 菜单

New Window新窗口	打开一个合当前窗口内容相同的新窗口。此菜单只有当前窗口是源文件窗口，汇编窗口，或者存储器窗口时才有效。
Cascade层叠	在主窗口内层叠显示窗口。
Tile平铺	在主窗口内平铺显示窗口。
Arrange Icons排列图标	重新排列主窗口中的图标。
Close All关闭所有	关闭主窗口外的其他窗口
Refresh刷新	更新当前窗口中显示的结果
Active激活	设置窗口为激活状态.
Static静止	设置窗口为静止状态.
(Open Window) 打开窗口	列出打开的窗口。 在图形边上显示有选中标志的窗口是当前窗口。选中一个窗口名字，被选中的窗口用作当前窗口。

(10)　[Help] 菜单

ID850QB Help帮助	显示帮助
Command Reference命令参考	打开"COMMAND REFERENCE"帮助窗口.
Main Window主窗口	显示主窗口的帮助
Current Window当前窗口	显示当前窗口的帮助
About... 关于	显示ID850QB的版本号。打开关于对话框。

工具栏

(1) 每个按钮的含义

工具栏上每个按钮的功能如下所述。当鼠标光标放置在工具栏的图标上，几秒后会弹出操作提示。

按钮	功能
Stop	停止用户程序的执行。 和[Run] 菜单 -> [Stop]功能相同
ReGo	CPU复位并且运行用户程序。 和[Run] 菜单 ->[Restart]功能相同
Go	CPU不复位，从当前PC指针处开始执行程序。 和[Run] 菜单-> [Go] 功能相同。
Go	忽略设置的断点，运行用户程序 和[Run] 菜单-> [Ignore break points and Go] 功能相同。
Ret	用户程序一直运行，直到返回 和[Run] 菜单- [Return Out] 功能相同。 注:此命令仅用在C语言函数
Step	单步执行(逐条执行程序指令) 如果一个函数或者子函数被调用 其指令逐条执行 和[Run] 菜单-> [Step In] 功能相同。
Ove	执行下一条指令(假定一个函数/调用声明为一步，执行程序)如果一个函数或者子函数被调用，其指令不被执行。
Res	CPU复位。和[Run] 菜单-> [CPU Reset]功能相同。
Open	打开显示文件加载对话框 和[File] 菜单-> [Open...]]功能相同。
Load	打开下载对话框。 和[File] 菜单-> [Download...]]功能相同。
Proj	打开工程文件加载对话框。 和[File] 菜单->[Project] -> [Open...]]功能相同。
Src	显示源文本。打开源文件窗口。 和[Browse] 菜单->[Source Text] ->功能相同。
Asm	显示反汇编结果，打开汇编窗口。 和[Browse] 菜单->[Assemble] ->功能相同。
Mem	显示存储器内容。打开存储器窗口。 和[Browse] 菜单->[Memory] ->功能相同。
Wch	显示观察内容。打开观察窗口。 和[Browse] 菜单->[Watch] ->功能相同。
Reg	显示寄存器内容，打开寄存器窗口。 和[Browse] 菜单->[Register] ->功能相同。

IOR	显示周边寄存器内容。打开IOR窗口 和[Browse] 菜单->[I/O Register] ->功能相同。
Loc	显示局部变量内容。打开局部变量窗口 和[Browse] 菜单->[Local Variable] ->功能相同。
Stk	显示堆栈跟踪结果。打开堆栈窗口。 和[Browse] 菜单->[Stack Trace] ->功能相同。
TrW [IECUBE]	显示跟踪结果。打开跟踪显示窗口。 和[Browse] 菜单->[Trace] ->功能相同。
Cov [IECUBE]	显示代码覆盖率测量结果。打开覆盖率窗口。 和[Browse] 菜单->[Code Coverage] ->功能相同。
Mgr	打开事件管理器 和[Browse] 菜单->[Event Manager...] ->功能相同。
Evn	登录并设置事件。打开事件对话框。 和[Event] 菜单->[Event...] ->功能相同。
Brk	登录并设置断点事件。打开断点对话框。 和[Event] 菜单->[Break...] ->功能相同。
Trc [IECUBE]	登录并设置跟踪事件。打开跟踪对话框。 和[Event] 菜单->[Trace...] ->功能相同。
Tim [IECUBE]	登录并设置定时器事件。打开定时器对话框。 和[Event] 菜单->[Timer...] ->功能相同。

(2) 工具栏的操作

工具栏显示与否，用户可以通过[Option]菜单中的 [Tool Bar].选项选择。

工具栏可以按以下两种方式显示。两种方式的显示通过选择调试器选项对话框来实现。

图 6-2： 工具栏 **(**图形方式**)**



图 6-3： 工具栏 **(**图形和文本方式**)**



窗口显示区域

这个区域显示不同的调试窗口。
显示窗口的大小可以被改变，也可将窗口最小化为图标形式。

状态栏

状态栏显示ID850QB和在线仿真器的状态.。当用户程序执行时，状态栏显示为红色。

工具栏显示与否，用户可以通过 [Option] 菜单> [Status Bar] 选项选择。

备注： 如果屏幕是的分辨率较低 (如800 - 600)，所有的状态栏上的状态都可能不显示。

图 6-4：状态栏

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
main.c#16	main	00000276	STOP	BREAK	Manual Break	AUTO	INS

(1) 程序名称	显示当前PC值所指示的程序文件名称。
源程序名称	显示当前PC值所指示的源文件名称。
行号	显示当前PC值所指示的行号。
(2) 函数名称	显示当前PC值所指示的函数名称
(3) PC值	显示当前PC值
(4) CPU状态	参考"表6-2：CPU状态".
(5) IE状态	参考"表6-3：IE状态". (如果有两个或多个状态，它们用'|' 定界进行显示)
(6) 断点原因	参考"Table 6-4 Break Cause".
(7) STEP模式	显示单步执行模式。显示从 [Option]菜单选择的下列模式 SRC:.源模式 INST:.指令模式 AUTO: .自动模式
(8) 键盘输入模式	显示键盘显示模式 INS：插入模式 OVR：覆盖模式 存储器窗口固定为覆盖模式

表 6-2 CPU 状态

显示	含义
HALT	停止模式
STOP/IDLE	软件停止模式，硬件停止模式，空闲模式
HOLD	总线保持模式
WAIT	等待模式
RESET	复位模式
POW OFF	目标系统没有加电

表 6-3：IE 状态

显示	含义
RUN	用户程序按顺序执行 (状态栏颜色改变)。
STEP	程序单步执行。
TRC	正在跟踪 **[IECUBE]**
TIM	定时器正在运行**[IECUBE]**
COV	覆盖率测量正在进行**[IECUBE]**
BREAK	发生了断点事件。
Time	显示用户程序从开始到断点发生事件的测量结果(运行终止时间)[注]
TIMER OVERFLOW	测量结果溢出

注： 可能的测量范围为20 ns(纳秒)到195.2 hours(小时)。(4K 分频) **[IECUBE]**
可能的测量范围为200 ns 到7 minutes(分钟)。 (DCK=10MHz) **[MINICUBE]**
可能的测量范围为100 us(微秒)到100 hours(小时)。 **[MINICUBE2]**

表 6-4：断点原因

显示	含义
人工断点	强制断点
临时断点	临时断点
软件断点	软件断点
跟踪溢出断点	跟踪溢出造成断点**[IECUBE]**
非映射断点	访问非映射区域 **[IECUBE]**
写入保护	企图写入写保护区域**[IECUBE]**
IOR 非法	对周边I/O寄存器非法访问**[IECUBE]**
定时器时间到断点	检测到运行时间到事件**[IECUBE]**
闪存宏功能	运行闪存宏功能**[MINICUBE]**
IRAM写保护(xxx xxx)	在断点期间，执行IRAM保护区域的校验检查并写入新的数值。**[IECUBE]** xxx xxx 指示相应的地址和数据(为避免重复，只有第一项被显示).
非法操作码陷阱	非法操作码陷阱断点 **[IECUBE]**
事件断点"事件名称或事件关联名称"	根据显示的事件名或事件关联名的事件原因停止

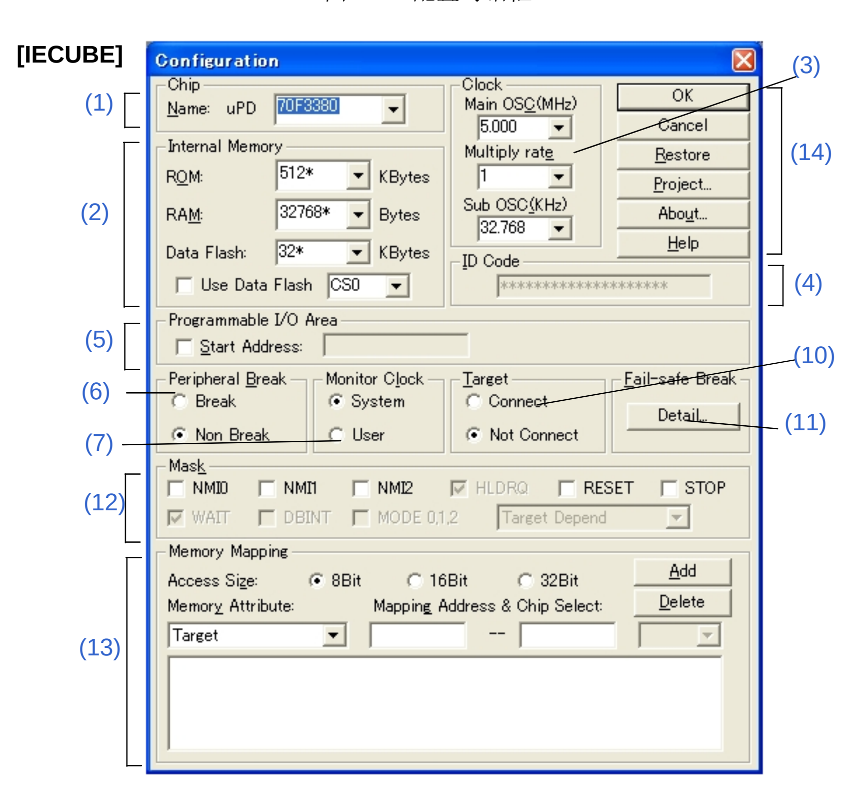
配置对话框

这个对话框用来实现ID850QB操作环境的显示和设置。 (参考"5.1 调试环境设置".)

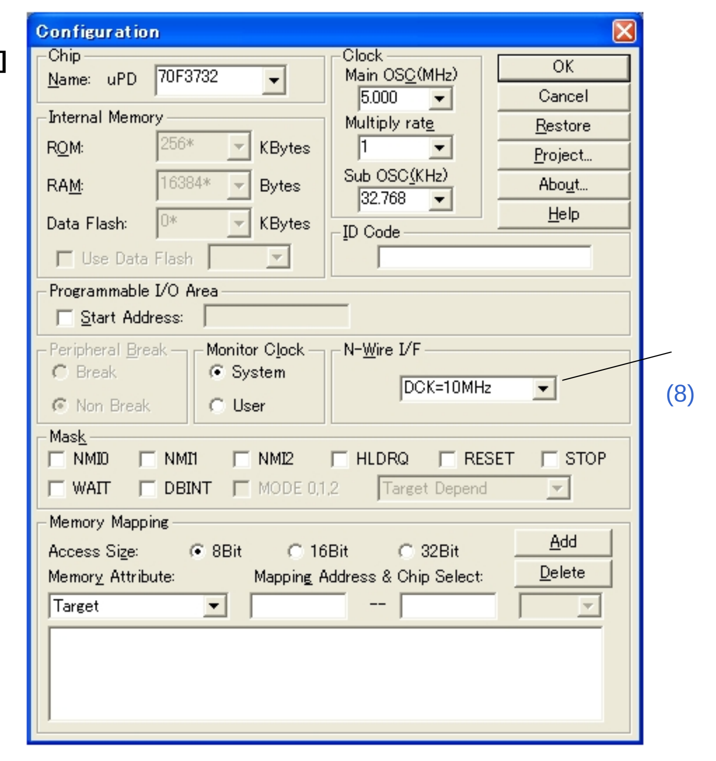
当ID850QB启动后，这个窗口自动显示。

但是，不同于必须读取工程文件才能显示对话框，对话框的设置不是必须的 (参考"5.15.1 调试环境 (工程文件)".)

图 6-5：配置对话框

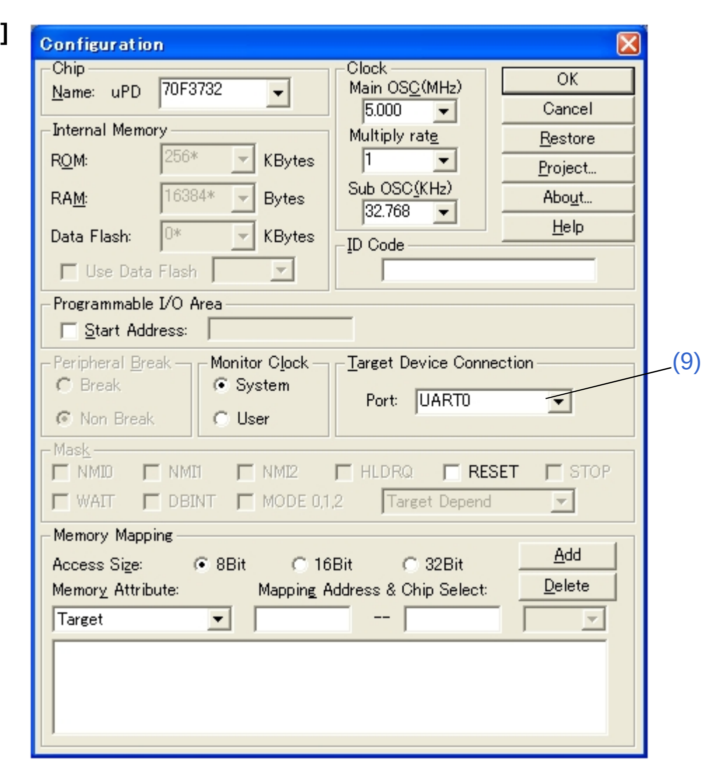


[MINICUBE]



(8)

[MINICUBE2]



(9)

- 打开
- 每个区域内容解释

打开

(当ID850QB 启动时，这个窗口自动打开。)
从 [Option] 菜单条选择[Configuration...] 选项。

每个区域内容解释

(1) 芯片

这个区域用来选择芯片名称。可以从下拉菜单中选择芯片名称
在下拉菜单区域，只有系统设备文件登录在登录表中的芯片名称才被显示。只用在调试器启动后才能指定这个区域。

备注: 在缺省情况下，上一次使用的芯片类型被显示。但是如果这个类型没有被登录登录，登录登录的第一个类型将被显示。

(2) 内部存储器

这个区域用来设置内部CPU存储区的大小。可以通过下拉式菜单或键盘输入。
缺省的值是通过选择"Chip"的设备文件得到并显示的(带有'*'的值)。

备注: 这个区域不能被选择。ROM和RAM的大小已经在设备文件中确定了。**[MINICUBE] [MINICUBE2]**

区域	含义	可设置的范围
ROM:	内部ROM大小设置	0, 8, 32, 64, 128, 256, 512, 1024 (KB)
RAM:	内部RAM大小设置	4096, 12288, 28672, 61440 (Bytes)
Data Flash:	设置数据闪存 (当使用的器件包含数据闪存时). 为了使用数据闪存区域，选择"Use Data Flash" 复选框。当使用V850ES系列时，在"Chip Select area"指定芯片选择(Chip Select)用来映射数据闪存	
Use Data Flash:		
Chip Select aera		

(3) 时钟

这项设置中可以设置系统主时钟和子时钟

Main OSC (MHz)主时钟频率	设定在主时钟倍频前的频率。 可以从下拉菜单选择频率(5.000, 8.000, 13.500, 或18.000) 或者直接输入
Multiply rate倍率	指定主时钟的工作时钟倍率。 这个值可以从下拉菜单中选择(1 to 10)，或者直接输入。
Sub OSC (KHz) 子时钟频率	指定子时钟频率 (当没有子时钟时这个参数是禁止选择的)。 可以从下拉菜单中选择或者直接输入

(4)　ID码 [MINICUBE] [MINICUBE2]

当ID850QB从内部ROM或者内部闪存读取代码时这个区域用来输入需要的ID码。(ID码认证)

输入20位十六进制的数字(10字节)作为ID码(缺省全"F")

ID码被存储在寄存器中，如果三次输入的ID码都失败，ID850QB会强制关闭。

备注1:　对于没有ROM或没有RCU(安全单元)的产品这个区域不用设置。
备注2:　对于更加详细的ID码的设定，参考N-Wire卡，MINICUBE或MINICUBE2用户手册

(5)　可编程I/O区域

这个区域用来指定可编程的I/O区域和起始地址

只有选择的芯片支持可编程I/O区域时，才可以指定可编程I/O区域的地址。当使用可编程I/O区域时，起始地址可以通过检查对话框进行输入。地址范围16KB对齐

备注:　当芯片的扩展I/O区域具有固定地址时，通过"Chip"选择芯片，系统将自动设定扩展I/O区域地址。

(6)　周边断点

这个区域用来指定在断点期间在线仿真器的周边仿真功能是否停止。

断点	停止
Non Break不产生断点	Not stopped不停止(缺省)

(7)　监视时钟[IECUBE]

这个区域用来指定监视程序的操作时钟在断点期间是否从子时钟切换到主时钟。

当一个产品没有子时钟时，这个区域不用设置

系统	操作时钟切换到主时钟，执行监视程序 (缺省). **注:** 在ID850QB中，通过操作PCC改变时钟，但是主时钟停止时不能改变。断点期间如果操作时钟切换到主时钟，当程序返回到用户程序执行时，时钟由返回到先前的设置。
用户	用户监视程序执行的时钟由用户程序指定

(8)　N-Wire I/F [MINICUBE]

这个区域用来选择N-Wire卡、MINICUBE提供给片上调试单元(DCU)的时钟。缺省情况下提供的是一个10MHz的时钟。

注意事项:　通常必须选择10MHz时钟。当选择20MHz时钟时ID850QB可能无法启动。

DCK=10MHz	DCK时钟是10 MHz (缺省方式下). **备注:** 在选择期间，执行时间测量功能所测量时间的最大值加倍，执行时间测量功能的分辨率减半。
DCK=20MHz	在这种方式下，DCK时钟是20 MHz.

(9)　目标设备连接[MINICUBE2]

这个区域用来选择连接MINICUBE2和目标系统的串行通信接口

MINICUBE2支持UART和CSI-H/S作为通信接口。

根据使用设备的情况，选择端口的类型可以变化。

(10)　目标[IECUBE]

这个区域用来选择目标系统是否和在线仿真器相连(参考"Table 3-2 错误消息输出模式 [IECUBE]")

连接	已连接
不连接	未连接

备注:　用来检测非法的电源状态。

缺省方式由检测目标系统的电源(TVDD)决定。

(11)　故障保险断点 [IECUBE]

这个区域用来选择**故障保险**断点功能。单击<Detail...> 按钮打开**故障保险**断点对话框可对**故障保险**断点的详细功能进行单独设置。

(12)　屏蔽

这个区域用来屏蔽从目标系统发来的信号

被屏蔽的管脚信号不输入到在线仿真器上。

屏蔽一个管脚只有在调试状态下目标系统不稳定时才执行。

备注1: 通过检测RESET，外部复位或内部看门狗定时器产生的内部复位都可以被屏蔽。这个时刻，内部复位能否被屏蔽取决于器件。

备注2:　如果使用支持TM标记的设备文件，选择模式00到模式1F(显示的模式由设备文件中的定义决定)。当IECUBE和目标设备都处于连接状态时，可以选择目标依赖。

(13)　存储器映射

这个区域用于设定映射。
通过"Access Size"选择访问存储器的大小，通过"存储器特性"设置映射特性，通过"映射地址"指定地址范围。
通过单击 <Add> 按钮设置存储器，相应的结果在窗口的底部显示。
可以映射的区域取决于产品的型号。

访问区域大小	选择访问存储器的大小。 用来指定 ID850QB 软件访问区间的大小；根据 MODE 管脚和 I/O 寄存器的设置来设置硬件的外部总线的操作方式	
	8Bit	使用ld.b和st.b指令访问存储器
	16Bit	使用ld.h指令和st.h指令访问存储器
	32Bit	使用ld.w指令和st.w 指令访问存储器
存储器特征	可以选择下列的映射属性。根据使用选择映射方式 (参考"表5-2 映射特性	
	Emulation ROM **[IECUBE]**	(使用存储器板) 选择在线仿真器代替ROM。映射单位为1MB。
	Emulation RAM **[IECUBE]**	(使用存储器板) 选择在线仿真器代替RAM。映射单位为1MB。
	Target	选择目标存储器，映射单位是1字节。
	Target ROM **[IECUBE]**	选择目标ROM。
	I/O Protect	选择I/O保护区域。 可以在目标区域指定的区域设置I/O保护区域。 因为在存储器窗口中设置的I/O保护区域被显示为"??"，这和非映射的区域类似，在存储器窗口不能自由进行读写操作，防止错误的读写操作此区域。映射单元是1字节。

Mapping Address & Chip Select[Note] **[IECUBE]映射地址和芯片选择**[注]	指定映射的地址。 通过键盘输入高端和低端的地址。 因为在仿真器内存中指定的这个区间范围为16 MB (16块1 MB的存储器), those areas 这些区间可以通过芯片选择功能分配给CS0到CS7决定的地址范围 (从下拉菜单列表中选择) 地址可以分配给任何1 MB区域。 可以通过一个 CS 指定多块存储器。

<Add>添加 <Delete>删除	这些按钮用来设置和删除映射。单击<Add>按钮，映射在指定区域中被设定，结果在窗口底部区域列出。从列表中选择项目单击，可删除该映射项。

注: 对于V850ES系列，芯片选择是固定的，或者没有芯片选择功能。当连接IECUBE时，如果扩展板没有安装，无此选择项。

注意事项1: 除非在IOR窗口和观察窗口中登录为I/O端口，这个被设置为 “I/O保护”的 区域不能被读取，只能在这些窗口中强行读取这个区域。

注意事项2: 如果外部存储器被映射，修改控制访问外部存储器的寄存器。(参考”5.1.4对控制访问外部存储器的寄存器进行修改”)

(14) 功能按钮

OK	使当前的环境生效。 设置环境并关闭此对话框 如果单击<OK>按钮后发生错误，ID850QB不能继续工作，并且被终止。
Cancel放弃	放弃修改并关闭对话框
Restore恢复	在对话框打开之前恢复以前的设置。
Project... 工程	打开工程文件加载对话框。如果在打开或者读取工程文件时发生错误，ID850QB不能继续工作，并且被终止。
About... 关于	打开关于对话框
Help帮助	显示当前窗口在线帮助文件

扩展选项对话框

这个对话框用来显示ID850QB的扩展选项的设置(参考"5.1 调试环境设置")

图 6-6：扩展选项对话框

Extended Option

(1)
Trace
Timetag Count Rate: 1
Memory Size: Min Max
Clear trace memory before run
Trace Data: Branch PC + Access Data
Mode: Speed Priority Trace Priority
Complement Data Add DMA Point

(2)
Timer
Count Rate: 1

(3)
RAM Monitor
Break When Readout: Whole IRAM + IOR Off
Redraw Interval: 500 msec

(4) On Mouse Click: Soft break Hard break
(5) Break Sound (6) Verify Check (7) Clear register when reset
(8) OK Cancel Restore Help

- 打开
- 每个区域解释

打开

选择[Option] 菜单中的[Extended Option ...] 选项.

每个区域的解释

(1)　跟踪[IECUBE]

这个区域用来设置跟踪功能（参考"5.10　跟踪功能　[IECUBE]"）(当IECUBE被连接时).

(a)时间标记速率计算:

这个区域用来在跟踪观察窗口中设置时间标记计数器的分频比。

分频比在下拉菜单中选择。

如果设置了分频比，时间标记计数器所必需的时钟数目会发生改变。

时间标记计数器分频比和最大测量时间的关系如下所示。

表 6-5：时间标记计数器分频比和最大测量时间的关系　(时间标记计数器(跟踪)

分频比	精度（ns)	可测量的最大时间	备注
TMCLK (1/1)	20	1.4 minutes分钟	时间标记计数器为32比特，外部时钟50MHz
TMCLK (1/2)	40	2.8 minutes分钟	
TMCLK (1/4)	80	5.7 minutes分钟	
TMCLK (1/8)	160	11.4 minutes分钟	
TMCLK (1/16)	320	22.8 minutes分钟	
TMCLK (1/32)	640	45.6 minutes分钟	
TMCLK (1/64)	1280	1.5 hours小时	
TMCLK (1/128)	2560	3 hours小时	
TMCLK (1/256)	5120	6 hours小时	
TMCLK (1/512)	10240	12.2 hours小时	
TMCLK (1/1024)	20480	24.4 hours小时	
TMCLK (1/2048)	40960	48.8 hours小时	
TMCLK (1/4096)	81920	97.6 hours小时	

(b)　存储器大小:

设置跟踪存储区的大小(缓冲区)。也就是，拖动knob按钮可以设置存储器大小。存储器大小可被设定为8K (最小),32K, 64K, 128K或256K (最大)。

注意事项: 设置的数值越大，记录的跟踪数据就越多。但是读取跟踪数据时反应相对变慢。

(c)　运行前清除跟踪存储器

程序运行前选中此选择框清除跟踪存储区。

(d)　跟踪数据:

选择需收集的跟踪数据

备注:　当选择[Options] 菜单中[RRM 功能]项后，设置和分支转移PC关联。

当选择[Option]菜单中[覆盖率功能]项后，设置和所有的PC关联。

表 6-6：跟踪数据采集和跟踪采集模式的关系

项目	跟踪范围			
	Branch PC分支起点和分支终点指令的PC值汇总。	**All PC**所有指令PC值汇总	**Access PC** 导致访问的指令PC值汇总	**Access Data** 访问地址和访问数据的汇总
Branch PC	跟踪	--	--	--
All PC	跟踪*a			--
Access Data	--	--	--	跟踪*b
Branch PC + Access Data	跟踪	--	--	跟踪*b
All PC + Access Data	跟踪*a			跟踪*b
Access Data + Access PC	--	--	跟踪	跟踪*b
Branch PC + Access Data + Access PC	跟踪	--	跟踪	跟踪*b

***a**　当包含"All PC"的跟踪数据被选中时，系统执行无条件跟踪模式。在这种情况下，不能同时设置限定跟踪模式和区域跟踪模式(条件跟踪模式设置是无效的。)

***b**　当包含"访问数据"的跟踪数据被选择时，当选中高速优先模式时(在(e)模式中选择速度优先)，当连续访问32次内部RAM后，数据有可能丢失。

(e)　模式:

指定跟踪数据的收集模式

备注:　当 [Option]菜单中[RRM Function/Coverage]被选择后，设置固定为速度优先。

速度优先	这个模式执行速度优先的跟踪模式 (实时操作). 在这种模式下，可能导致某些数据的丢失 (参考"表6-6 跟踪数据采集修和跟踪采集模式的关系".)
跟踪优先	执行采集数据优先的跟踪模式 (非实时操作). 因为，为了可靠地采集跟踪数据，当数据可能丢失时，CPU执行流水线可能瞬间停止，将会丧失用户程序的实时特性。 在(d)跟踪数据中如果选择分支PC或全部PC时，不能选择跟踪优先级模式。

(f)　反白数据

选择复选框执行跟踪数据的反白显示(缺省：选中)

(g)　增加DMA传输点

选择复选框可以执行DMA传输点跟踪功能(缺省情况下复选框被选中)。

当复选框被选择时，DMA的起始帧和结束帧被标记出来。

(2)　定时器 [IECUBE]

设置定时计数器的速率。

在定时器对话框中计数速率 "Count Rate"区显示计数速率。

速率值可通过下拉菜单设置。

定时器分频比和最大测量时间的关系如下。

表 6-7：定时器分频比和最大测量事件之间的关系(定时计数器(定时器))

分频比	分辨率 (ns)	最大测量时间	备注
TMCLK (1/1)	20	2.8 minutes分钟	在50 MHz外部时钟方式下，定时器为33 bits,
TMCLK (1/2)	40	5.7 minutes分钟	
TMCLK (1/4)	80	11.4 minutes分钟	
TMCLK (1/8)	160	22.8 minutes分钟	
TMCLK (1/16)	320	45.6 minutes分钟	
TMCLK (1/32)	640	1.5 hours小时	
TMCLK (1/64)	1280	3 hours小时	
TMCLK (1/128)	2560	6 hours小时	
TMCLK (1/256)	5120	12.2 hours小时	
TMCLK (1/512)	10240	24.4 hours小时	
TMCLK (1/1024)	20480	48.8 hours小时	
TMCLK (1/2048)	40960	97.6 hours小时	
TMCLK (1/4096)	81920	195.2 hours小时	

(3) RAM 监视器

(a) 读出时中断

当用户程序产生即时断点时，选择这一项以指定RAM的采样范围(参考"5.13.2 伪实时监视功能 (读出时产生断点)"。)

Whole整个	整个存储器空间备注 **备注：** 当大量的窗口打开时，因为被读取的内存范围比较宽，用户程序的执行将停止一段较长的时间，因为需要读出的地址范围太大了。
IRAM + IOR	内部RAM区IOR区
Off	禁止伪实时监视功能 (读出时产生断点)(缺省)。

注: RRM设置对话框指定的范围不包括在内。

(b) 刷新间隔:

指定RAM采样的采样时间(ms)。

可以指定从0到65500个100 ms间隔。

如果设定0，或者这个区域是空的，不能实时显示数据。

(4) 关于鼠标点击:

这个区域指定缺省断点是软件断点还是硬件断点。在源文件窗口或汇编窗口中指定的区域点击鼠标按钮可以在标记处设置一个断点 (参考"5.4.2 断点设置").

软断点	设置软件断点
硬断点	设置硬件断点

(5) 断点声音

如果选中了复选框，当断点发生时蜂鸣器发出声音。

(6) 校验检查

这个区域用来指定是否对写入存储器的数据进行校验检查。

当执行下载、存储器填充和存储器复制操作时，系统执行校验操作。当一个变量或数据在观察窗口或存储器窗口修改后并写入存储器时，也执行校验检查操作。

注意事项: 在写入内部闪存时 (包括下载), 无论这个区域的复选框是否被选择，都不执行校验检查操作，而闪存自写入的内部校验总是执行的 (不执行读校验)。**[MINICUBE]**

(7) 复位时清除寄存器

选择此复选框用来在CPU复位情况下清除程序寄存器(从r1到r31)和EIPC, EIPSW, FEPC, FEPSW, CTPC, CTPSW, and CTBP 等寄存器。

在缺省方式下，寄存器内容不被清除。

(8) 功能按钮

OK	使设置生效并关闭对话框
Cancel取消	放弃修改并关闭对话框
Restore恢复	在对话框打开时，恢复以前的设置
Help帮助	显示当前对话框在线帮助文件

故障保险断点对话框

[IECUBE]

这个对话框用来进行**故障保险**断点设置(参考"5.4.5　故障保险(Fail-safe)断点功能 [IECUBE]".)

当读取一个工程文件时，读取工程文件的结果将在这个对话框中显示。

备注:　　对于更加详细的**故障保险**断点功能，参考在线仿真器和仿真板的用户手册。

图 6-7：故障保险断点对话框

- 打开
- 每个区域内容解释

打开

在配置对话框中单击<Detail...> 按钮。

每个区域解释

(1) 保护

这个区域执行**故障保险**断点保护设置功能。

复选框中设置的**故障保险**断点都是处于保护状态。在缺省设置下，所有的复选框都被选择。

内部ROM	这个区域用来实现对内部ROM区域的保护设置	
	非映射	访问禁止访问的区域
	写	写入禁止写入的区域
内部RAM	这个区域用来实现对内部RAM区域的保护设置	
	非映射	访问禁止访问的区域

I/O 寄存器	这个区域用来实现对周边I/O寄存器区域保护设置	
	非映射	访问禁止访问的区域
	读	读取禁止读取的区域
	写	写入禁止写入的区域
外部存储器	这个区域用来实现对外部存储器区域保护设置	
	非映射	访问禁止访问的区域
	写	写入禁止写入的区域

(2)　校验

这个区域用来检验**故障保险**断点。

当访问复选框选中的项目时，系统执行校验检查

内部RAM	这个区域用来实现对内部RAM区域的校验检查设置	
	非映射写	写入禁止写入的区域

(3)　功能按钮

OK	使设置生效并关闭对话框
Cancel取消	放弃修改并关闭对话框
Help帮助	显示此对话框在线帮助文件

RRM 设置对话框

[IECUBE]

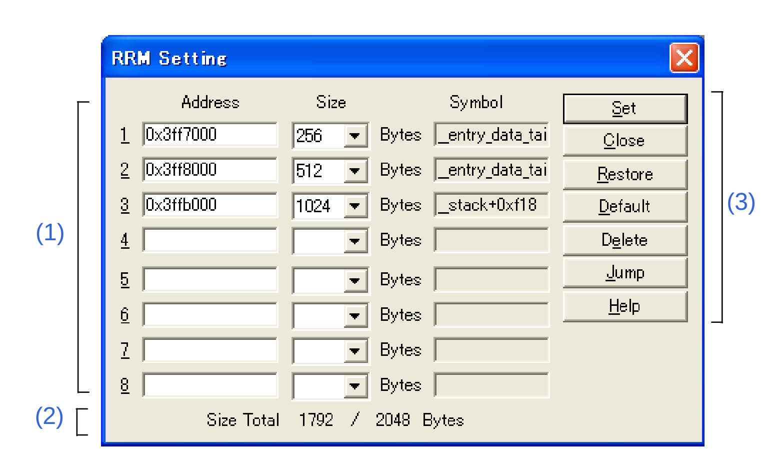
这个对话框用来设置RRM功能的采样范围。(参考"5.13 RRM 功能")。

最多可以设置8个以256字节为单位的采样区域。

为8个点指定的总的区域大小不能超过2048字节

注意事项: 当在[Option]中选择[RRM] 以外的功能时，系统无法打开这个对话框

图 6-8：RRM 设置对话框



- 打开
- 每个区域解释

打开

这个对话框因打开方式不同设置也不相同。

(a) 当从RRM对话框进行设置时

通过选择 [Option] 菜单的[RRM Setting...]选项打开对话框。在这种方式下，"地址"和"大小"中的数据是手工输入的。

(b) 当通过存储器窗口进行设置时

通过打开存储窗口打开此对话框，在窗口中选择一个地址，然后从上下文菜单选择[RRM Setting...]。这种情况下，选择的地址在"Address"的空行上显示，在"Size"的空行上显示"256"，在"Symbol"空行上显示地址到符号转换得到的值

备注: 如果8个设置点的大小总和已超过2048字节，可以打开这个对话框，但是不能设置数值

(c)　当从观察窗口进行设置时

这个对话框时从观察窗口打开的，在窗口中选择一个变量，然后从上下文菜单选择 [RRM Setting...]。

这种情况下，变量到地址转换得到的值在"Address"的空行上显示，在"Size"的空行上显示"256"，在"Symbol".空行上显示变量到符号转换得到的值

备注:　如果8个设置点的大小的总和超过2048字节，这个对话框可以被打开，但是不能设置数值

每个区域解释

(1)　采样范围设定区域

Address地址	这个区域用来指定RRM功能中采样的起始地址 缺省的数值基数为16进制。一个地址也可用符号或者表达式指定(参考"表5-6 指定符号".) 输入以后单击<Set>按钮使输入有效。
Size大小	这个区域用来指定"Address"采样范围。 这个值可以被选择伪256, 512, 768, 1024, 1280, 1536, 1792, 2048. 但是，8个位置指定的数值总和不能超过2048字节
Symbol符号	这个区域显示在"Address"中指定的地址符号 指定的地址显示为一个符号或符号加偏移量。 如果地址没有被设置，什么也不显示。

备注:　当设置允许时，　地址以256字节为单位对齐，但是重复地址是不允许的。

(2)　总的数值

这个区域显示在"Size"中指定的总的数值，如果总的数值超过2048字节，它被显示为红色。

(3)　功能按钮

Set　　设置	确定指定的采样范围
Close关闭	关闭对话框
Restore恢复	恢复对话框打开之前的设置
Default缺省	清除当前设置，并设置内部RAM起始地址为第一行地址，同时第一行大小设置为"2048"　。

删除	删除光标处设置的数值
Jump跳转	打开存储器窗口， 在"Address" 中光标处显示地址。 当存储器窗口是激活窗台时，可以执行跳转功能。 如果同时打开多个存储器窗口，这些窗口必须设置为静止状态 (参考"5.16.1 激活状态和静止状态".)
Help帮助	显示对话框在线帮助文件

闪存选项对话框

[IECUBE]

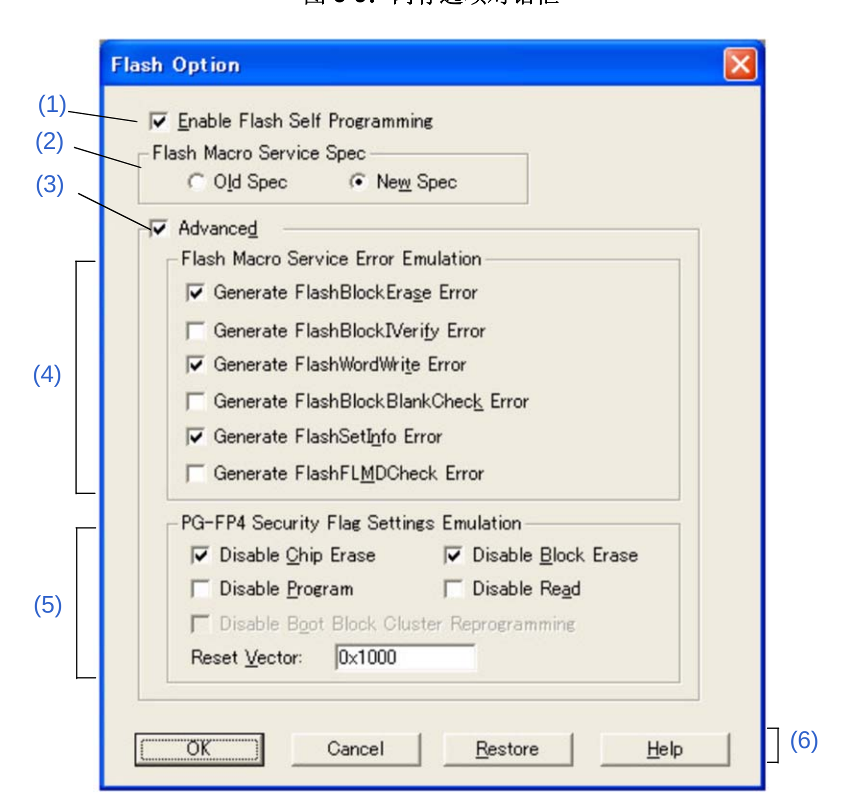
这个对话框用来实现对闪存自编程仿真功能的设置(代码闪存)。

在用户程序执行期间，无法打开这个对话框。

对于列表中支持的器件，可用闪存仿真功能列表，和注意事项,参考"闪存自编程仿真的特殊注意事项"。

注意事项： 对于没有闪存的设备，系统不能打开这个对话框

图 **6-9**：闪存选项对话框



- 打开
- 每个区域解释
- 闪存自编程仿真的特殊注意事项

打开

选择[Option]菜单的 [Flash Option...].选项

每个区域的解释

(1) 闪存自编程的使能

如果选择此项，将使能自编程仿真功能。

使能(3)高级设置。缺省方式下系统不选择这个条目。

(2) 闪存的宏功能说明

当闪存自编程仿真功能处于使能状态，在这个区域选择闪存指定的宏功能业务。

只有当闪存按"类型1"操作时这个区域才是合法的
如果选择"新规范"(New Spec)，"禁止读"和"复位矢量"在(5) PG-FP4安全标志设置仿真中变为可选择的。在复位向量地址的"复位向量"域内的内部闪存的地址范围内输入一个地址。

Old Spec旧规范	闪存自编程Ver. 2.00或早期版本 (缺省)
New Spec新规范	闪存自编程Ver. 3.00或更新版本

(3) 高级

如果选择这些项，闪存自身编程仿真功能的详细设置处于使能状态。这些使能的设置是 (4) 闪存宏功能错误仿真和(5) PG-FP4安全标志设置仿真。 缺省方式下这些项是不选的。

(4) 闪存宏功能错误仿真

这些项目设置自身的库函数操作。

当选中此复选框时，如闪存损坏，强制返回一个错误值 (当闪存损坏时，返回的错误值在正常仿真情况下是不能返回的)

可设置的条目如下所示，缺省方式下，所有项目都是不选的。

产生 Flash Block Erase 错误	从FlashBlockErace函数中返回错误值
产生 FlashBlocklVerify 错误	从FlashBlockIVerify 函数中返回错误值
产生 FlashWordWrite 错误	从FlashWordWrite 函数中返回错误值
产生 FlashBlockBlankCheck 错误	从FlashBlockBlankCheck函数中返回错误值
产生 FlashSetInfo 错误	从FlashSetInfo函数中返回错误值
产生 FlashFLMDCheck 错误[注]	从FlashFLMDCheck函数中返回错误值

注:　IECUBE不能读取FLMD0管脚上的值 (通常，假设FLMD0管脚输入为高电平，FlashFLMDCheck 函数总是返回一个正常结束)

当发生FLMD0输入低电平错误需要在FlashFLMDCheck函数中强制返回错误消息时，选中"GenerateFlashFLMDCCheck Error" 复选框。

(5)　PG-FP4 安全标志设置仿真

当使用闪存编程器PG-FP4并设置了闪存的安全标志时，安全标志的初始值可以仿真得到。

可设置的项目如下：缺省方式下，所有的项目都是不选的。

Disable Chip Erase禁止芯片擦除	禁止芯片擦除
Disable Block Erase禁止块擦除	禁止块擦除
Disable Program禁止编程	禁止写
Disable Read注1 禁止读	禁止读
Disable Boot Block Cluster Reprogramming注2禁止重新写入引导区	禁止重新写入引导区
Reset Vector: 注1复位矢量	复位矢量地址设置

注1:　这一项只有当闪存工作在"类型1"时才能被选择，而且闪存自身编程版本为3.00或者以后版本。
注2:　只有当闪存工作在"类型2"和"类型3"时，这一项才能被选择。

(6)　功能按钮

OK	使设置有效并关闭对话框
Cancel取消	关闭此对话框
Restore恢复	对话框打开之前恢复以前的设置
Help帮助	显示对话框在线帮助文件

闪存自编程仿真的特殊注意事项

(1)　支持设备列表

下面列出2006年2月以后支持闪存自身编程仿真功能的器件

表 6-8：闪存自编程仿真支持器件

闪存类型	器件
Type1类型1	V850ES/Sx2, V850ES/Fx2, V850E/RS1, V850E/IA4, μPD70F3229Y, V850ES/Hx2, V850ES/Jx2
Type3类型3	V850ES/Kx1 (只对具有片上单电源供电的闪存微控制器) , V850ES/Kx1+, V850ES/Kx2

(2)　可以使用的闪存仿真功能列表

下表给出闪存功能是否能被仿真和ID850QB中闪存自身编程仿真功能的限制(仿真：可以被仿真；限制：可以被仿真但有许多限制；不能仿真：不能被仿真)

表 6-9：可执行的闪存仿真功能列表(类型 1)

闪存功能	功能简述和限制	能否仿真
Type1 (V850ES/Sx2, V850ES/Fx2, V850E/RS1, V850E/IA4, μPD70F3229Y, V850ES/Hx2, V850ES/Jx2)		
FlashEnv	闪存环境初始化/终止功能	仿真
FlashBlockErase	单块擦除功能	仿真
FlashWordWrite	单字写入功能 **限制:** 如果一个监视区的地址指定为第三个参数，在不可预知的地址产生故障保险断点。	限制
FlashBlockIVerify	单块内部校验处理功能	仿真
FlashBlockBlankCheck	单块查空功能	仿真

闪存功能	功能描述和限制		能否仿真
FlashGetInfo	闪存信息获得功能		
	Option = 2	CPU编号和闪存总数 **限制:** 在配置对话框中设置的器件名 (4位数) 作为CPU编号返回	限制
	Option = 3	安全信息	仿真
	Option = 4	获得引导区交换信息 **限制:** 不能反映引导区交换信息	限制
	Option = 5 + 块号	获取块的结束地址	仿真
FlashSetInfo	闪存信息设置功能。 **限制:** 引导区交换设置被忽略。		限制
FlashStatusCheck	检查最近使用过的闪存操作状态的功能。 **限制:** 对于FlashBlockErase和FlashBlockBlankCheck， 返回值从FE_BUSY切换到FE_OK的时间与实际设备中是不同的。		限制
FlashBootSwap	引导区交换功能		不能仿真
FlashSetUserHandler	用户中断管理登录功能		仿真
FlashFLMDCheck	FLMD0管脚状态检查功能		仿真
FlashSetInfoEx	闪存信息设置功能 **限制:** 引导区交换设置被忽略		限制 **注**
FlashNWordRead	多字读取功能 **限制:** I如果保护区中某个地址被指定为第三个参数，将在未知地址发生一个故障保险断点事件。		限制 **注**

注:　　这个功能在Ver.5.00及以后的版本中增加到闪存自编程中。

表 6-10：可执行的闪存仿真功能列表 (类型 3)

<table>
<tr><th>闪存功能</th><th colspan="2">功能概述及限制</th><th>能否仿真</th></tr>
<tr><td colspan="4">Type3 (V850ES/Kx1(只对具有片上单电源供电的闪存的微控制器, V850ES/ Kx1+, V850ES/Kx2)类型 3(V850ES/Kx1(), V850ES/ Kx1+, V850ES/Kx2)</td></tr>
<tr><td>FlashEnv</td><td colspan="2">闪存环境初始化/结束功能</td><td>仿真</td></tr>
<tr><td>FlashBlockErase</td><td colspan="2">单个块擦除功能</td><td>仿真</td></tr>
<tr><td>FlashWordWrite</td><td colspan="2">单字写功能
限制:
如果保护区中某个地址被指定为第三个参数，在未知地址发生一个故障保险断点事件。</td><td>限制</td></tr>
<tr><td>FlashBlockIVerify</td><td colspan="2">单个块内部校验功能</td><td>仿真</td></tr>
<tr><td>FlashBlock-BlankCheck</td><td colspan="2">单个块查空功能</td><td>仿真</td></tr>
<tr><td rowspan="5">FlashGetInfo</td><td colspan="3">闪存信息获取功能</td></tr>
<tr><td>Option = 2</td><td>CPU编号和闪存总数
限制:
设备文件名字(四位数)作为CPU号被返回.</td><td>限制</td></tr>
<tr><td>Option = 3</td><td>安全信息</td><td>仿真</td></tr>
<tr><td>Option = 4</td><td>获取引导区交换信息
限制:
引导区交换信息</td><td>限制</td></tr>
<tr><td>Option = 5 + 块号</td><td>获取块的结束地址</td><td>仿真</td></tr>
<tr><td>FlashSetInfo</td><td colspan="2">闪存信息设置功能
限制:
引导区交换设置被忽略。</td><td>限制</td></tr>
<tr><td>FlashBootSwap</td><td colspan="2">引导区交换功能</td><td>不仿真</td></tr>
<tr><td>FlashFLMDCheck</td><td colspan="2">FLMD0管脚状态检查功能</td><td>仿真</td></tr>
<tr><td>FlashWordRead</td><td colspan="2">读数据功能
限制:
如果保护区中某个地址被指定为第三方参数，在未知地址发生一个故障保险断点事件。</td><td>限制</td></tr>
<tr><td>FlashIVerify</td><td colspan="2">内部校验功能(EEPROM)</td><td>不仿真</td></tr>
<tr><td>FlashBlankCheck</td><td colspan="2">查空功能(EEPROM)</td><td>不仿真</td></tr>
<tr><td>EEPROM_Init</td><td colspan="2">EEPROM区初始化功能(EEPROM)</td><td>不仿真</td></tr>
<tr><td>EEPROM_Write</td><td colspan="2">EEPROM写功能(EEPROM)</td><td>不仿真</td></tr>
</table>

闪存功能	功能概述及限制	能否仿真
EEPROM_Read EEPROM读	EEPROM读功能(EEPROM)	不仿真
EEPROM_Copy EEPROM复制	EEPROM复制功能(EEPROM)	不仿真
EEPROM_VChK	EEPROM有效区域检查功能(EEPROM)	不仿真
EEPROM_Erase EEPROM擦除	EEPROM擦除功能(EEPROM)	不仿真

(3)　注意事项

执行闪存自编程时的注意事项如下所述

编号.	描述
1	下列情况闪存自编程仿真功能无效。 (a)　内部ROM大小没有设为缺省值 解决方案：在设置对话框中内部ROM大小设为缺省值。 (b)　如果运行前使用了两个断点 解决方案：运行前禁止或者删除其中一个断点。
2	当闪存自编程仿真功能允许，调试功能有以下限制。 (a)　内部ROM 和内部RAM的大小无法改变。 (b)　DMM和伪RRM功能无效。 (c)　如果SP寄存器值为0(没有指向内部RAM)将产生非法断点。 在寄存器值初始化并指向一个相应位置(比如内部RAM)之前，如果发生一个类似事件的断点,将在堆栈区发生一个非法断点。如果在此时期有发生这样一个断点的可能，请在程序执行前为SP设置一个相应值。 (d) 如果在扩展选项对话框中选中"Clear register when reset"，闪存宏功能改变的寄存器R3值在仿真器复位过程中被清除。 (e)　如果下面所示的限制应用到IECUBE，将产生一个非法断点。清除故障保险断点对话框。 -如果下面所示的限制应用到IECUBE，将产生一个非法断点。
3	当自编程仿真功能被允许后，从0地址开始的4字节区域被保留，在0地址处写入一个4字节指令jr 0xfffd6。因此，当在复位向量地址0处使用此功能时，在地址4处开始的区域分配一个启动程序。 当自编程仿真功能被禁止后，从0地址开始的4字节区域被请零。代码内容中不能有分支指向地址0，即使此功能用作复位向量地址0。 为了在实际使用器件时执行同仿真时相同的程序，推荐使用下面的语句。 #RESET handler(in the case of address 0) .section　　"RESET", text jr　　__start　　--Overwitten by jr 0xfffd6 jr　　__start

编号	描述
4	如果地址0被指定为复位向量地址，复位向量设在地址4。如果指定不同于地址0的地址，指定的地址被设为复位向量而不用加4。
5	关于仿真过程中FlashBlockErase()和FlashBlockBlankCheck()后的FlashStatusCheck()操作，FlashStatusCheck()的返回值从FE_BUSY到FE_OK发生变化的时间与实际器件中的不同。
6	如果地址被指定为位于被保护区域FlashWordWrite，FlashWordRead或FlashNWordRead的第三个参数，那么将访问非法的存储区地址，并且在未知地址发生故障保险断点事件。 修正FlashWordWrite，FlashWordRead，或者 FlashNWordRead之一相应的地址。
7	要使能闪存操作对话框中所作的设置，一定要复位CPU并重新执行程序；否则，设置不会起作用。
8	为调试器工作区保留至少84(54H)字节的堆栈区域。 在闪存写入仿真期间，当断点事件发生时，调试器占用一个最小84(54H)字节大小的区域。当中断允许时，另外84(54H)字节的堆栈区域被用作调试器工作区。 如果允许多个中断，每级中断必须有84(54H)字节作为堆栈区域。
9	CPU复位后，内部RAM中的数据被破坏。正常情况下，复位后实际器件内部RAM中的数据不能确定，但是注意操作可能不同。
10	如果不正确的调用闪存功能，或者调用一个不支持的闪存功能，将返回"1"。

数据闪存选项对话框

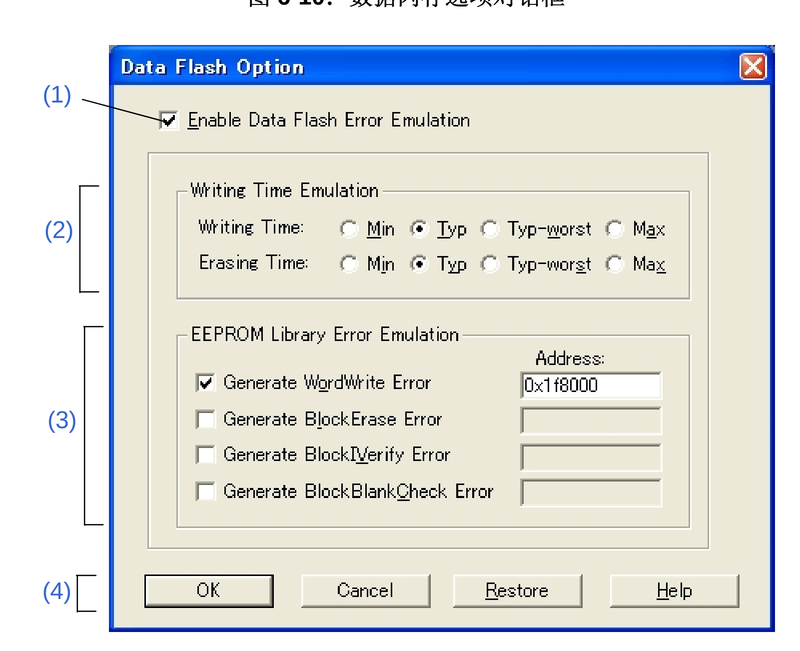
[IECUBE]

这个对话框用来对数据闪存错误仿真功能进行设置 (数据闪存)

在用户程序执行期间这个对话框无法打开

注意事项: 当使用的器件没有数据闪存时，系统无法打开这个对话框

图 6-10：数据闪存选项对话框



- 打开
- 每个区域解释
- 注意事项

打开

选择 [Option] 菜单中[Data Flash Option...]选项.

每个区域的解释

(1) 使能数据闪存错误仿真功能

如果选择这一条目，将使能数据闪存错误仿真功能。这将使能(2) 写入时间仿真和 (3) EEPROM 库错误仿真
缺省方式下不选择这一条目。

(2) 写入时间仿真

这些条目指定写入或删除数据闪存内容的时间。

从下列表中选择写入时间 (写入时间) 和删除时间 (删除时间)

Min最少	不进行重试
Typ典型	闪存宏指定的典型次数(缺省)
Typ-worst典型最坏	闪存宏指定的最大次数
Max最大	指定的重试最大次数

(3) EEPROM库错误仿真

这些条目设置EEPROM库函数的操作方式。

当选择复选框时，正常的仿真下不返回的错误值可被强制返回。

选择相应的复选框，可以指定一个数据闪存中的地址，在此地址处可产生一个确定的错误。

可设置的项目如下：在缺省方式下，将不选择所有的项

产生字写入错误	返回字写入函数的错误值
产生块删除错误	返回字块删除函数的错误值
产生块校验错误	返回字块校验函数的错误值
产生块空白校验错误	返回字块空白校验函数的错误值

(4) 功能按钮

OK	使设置生效关闭对话框
Cancel取消	关闭对话框
Restore恢复	恢复对话框打开之前的设置.
Help帮助	显示对话框在线帮助文件

注意事项

(1) 如果在配置对话框中没有选择”使用数据闪存”复选框，系统无法打开数据闪存选项对话框。

(2) 如果在配置对话框中选择”使能数据闪存错误仿真” 复选框，同时又清除”使用数据闪存”复选框，系统无法使用数据闪存选项对话框

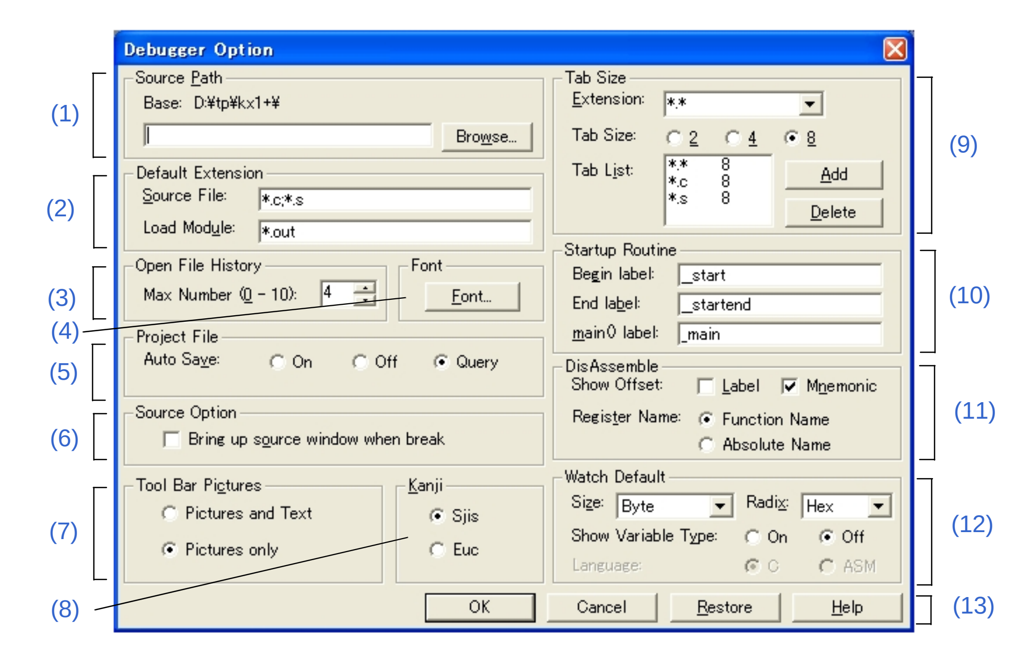
(3) 如果在RRM设置对话框中将一个RRM区域设置为数据闪存，在用户程序执行期间写入的数据将无法反显，因为写入数据闪存是通过库Library进行的。基于同样的原因，通过存储器窗口访问监视器功能(读/写，读写)，写入的数值也不能高亮反显。

(4) 在用户程序执行期间，禁止通过DMM对话框写入数据闪存。

调试选项对话框

这个对话框用来显示和设置ID850QB的各种选项

图 6-11：调试选项对话框



- 打开
- 每个区域的解释

打开

选择[Option]菜单中的[Debugger Option...]选项.

每个区域的解释

(1)　源路径

这个区域用来指定源文件或文本文件的搜索路径



Base:	此路径是相对路径的基础。按下面次序决定根目录。 **1)**工程文件加载路径 **2)**最近的加载模块或者hex文件的加载路径 **3)** Windows当前路径
Text box文本框	此区域用来指定搜索路径。 指定路径方法，或者直接在文本框输入，或者单击<Browse...>按钮。也可以指定相对路径。 单击<Browse...>按钮打开[Add Source path添加源路经]对话框。使用";" (分号) or "," (逗号)分隔路径。 **图 6-12 [Add Source path 添加源路经] 对话框**

备注1:　可以指定包含";"和/或","的源路径。不存在的目录不能被指定。

备注2:　当对话框打开后，根目录立即被选择和打开。如果选择的路径已经是源路径，将不增加该路径

备注3:　源路径长度最多可有4,095字符，包括扩展名前面的点号(.)。

如果指定源路径的字符串超过4096个字符，有效的源路径使前4095个字符的内容，后面的字符被忽略。

(2)　缺省扩展

这个区域用来指定缺省扩展。

扩展的定界采用" " (空格),";" (分号) 或 "," (逗号).

Source File:源文件	打开浏览对话框，选择[File]菜单-> [Open...]设定源文件的扩展名。 缺省的扩展名为" *.c, *.s"。
Load Module:加载模块	当下载对话框打开后，设置显示的加载模块的扩展名。 缺省的扩展名为" *.out".

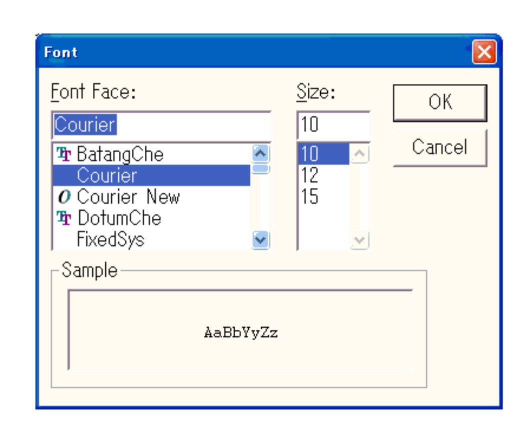
(3)　打开文件历史

这个区域用来设置[File]菜单下部打开文件的历史记录数目。缺省为4。如果指定0，这个菜单不显示历史记录

(4)　字体

这个区域用来指定源文件窗口、观察窗口、快速观察对话框、局部变量窗口和堆栈窗口中显示的字体。单击 <Font...> 按钮打开字体对话框，可以看到可设置的字体和大小。

图 6-13：[字体]对话框



(5)　工程文件

这个区域用来设定自动保存的工程文件. (参考"5.15.1 调试器环境 (工程文件)".)

<table>
<tr><td rowspan="4">自动保存</td><td colspan="2">设置ID850QB 终止时是否自动保存工程文件。</td></tr>
<tr><td>On</td><td>自动保存工程文件</td></tr>
<tr><td>Off</td><td>不自动保存工程文件</td></tr>
<tr><td>Query</td><td>在ID850QB终止时，显示退出调试器对话框(缺省)</td></tr>
</table>

(6)　源选项

这个区域用来设置断点处源文件窗口的操作方式。

通过选择该复选框，断点处激活的源文件窗口将在前面显示。

如果没有激活的源文件窗口，或者加载的模块文件中不包含调试信息，系统显示一个激活的汇编窗口。

(7)　工具栏图形

这个区域设置在工具栏上显示的按钮。(参考"(2) 工具栏的操作".)

图形和文本方式	按钮的图象和字符同时显示
图形方式	用图形方式显示按钮(缺省)

(8)　日文汉字

在这个区域不能选择日文汉字。

(9)　制表符数值

这个区域用来为每个可显示的扩展名类型设置制表符的宽度。

Extension:扩展	设置一个扩展名。从键盘设置一个扩展名，或者从下拉菜单中选择一个。
Tab Size:制表符宽度	选择制表符宽度。选择多少空格作为tab码显示。
Tab List制表符列表:	对每个扩展名显示制表符的大小
<Add添加>	选择"Extension扩展名:" 和 "Tab Size制表符宽度:"，单击此按钮改变制表符宽度设置。
<Delete删除>	在"Tab List: 制表符列表"中选中此项并单击此按钮，删除制表符宽度设定。

(10)　启动例程

这个区域用来指定首地址、尾地址并且显示起始程序的文本区的起始符号 (代码区域)

通过指定这个区域，通过在下载对话框加载模块格式目标文件，可以立即打开源文件 (如果这时PC位于其实标号和终止标号之间，ID850QB显示的代码起始地址时从"main()"标号开始的).

开始标号	指定第一个符号的地址 (缺省：_start)
终止标号	指定第一个符号的地址 (缺省： _startend)
Main() 标号	指定显示的其实符号(缺省：_main)

注意事项1:　如果指定的符号不正确，在PC到达源文件对应的地址之前，源文件不能打开。而且单步执行时不能跳过启动程序

注意事项2:　请确认指定的区域。如果这个区域是空的，对话框不能关闭。

(11) 反汇编

这个区域用来设置反汇编显示。

<table>
<tr><td rowspan="3">显示偏移量</td><td colspan="2">指定在反汇编显示时是否显示一个偏移量(符号 + 偏移量)。
当偏移量没有显示时，只有与数字值符合的符号才被显示。如果没找到符合的符号，数字值作为十六进制数显示，不会改变。</td></tr>
<tr><td>标号</td><td>指定在标号域是否显示偏移量，缺省条件下，不显示偏移量。</td></tr>
<tr><td>助记符</td><td>指定在助记符域是否显示偏移量，缺省条件下，显示偏移量。</td></tr>
<tr><td rowspan="3">寄存器名称</td><td colspan="2">在反汇编显示时，此区域用来选择寄存器助记符名称的显示方法。</td></tr>
<tr><td>功能名称</td><td>使用功能名称或者代号显示寄存器名字。(缺省)</td></tr>
<tr><td>绝对名称</td><td>使用绝对名称显示寄存器名字。</td></tr>
</table>

(12) 缺省观察

这个区域用来设置观察窗口中可以观察的符号。

<table>
<tr><td>大小</td><td colspan="2">如果下拉菜单中指定了[Adaptive]，选择数据的缺省显示长度。</td></tr>
<tr><td rowspan="6">基数</td><td colspan="2">如果在下拉菜单列表中指定了[Proper]选项，将设定输据显示的缺省基数。
此列表中选中的项目也反映在观察数据的subscript上，例如观察窗口中的数组变量(或者标号)(缺省为十六进制)，当设置改变以后登录在观察窗口。</td></tr>
<tr><td>Hex</td><td>显示十六进制数据 (缺省)</td></tr>
<tr><td>Dec</td><td>显示十进制数据</td></tr>
<tr><td>Oct</td><td>显示八进制数据</td></tr>
<tr><td>Bin</td><td>显示二进制数据</td></tr>
<tr><td>String</td><td>显示字符串数据</td></tr>
<tr><td rowspan="3">显示变量类型</td><td colspan="2">选择显示/不显示指定的变量类型</td></tr>
<tr><td>On</td><td>显示变量的类型.</td></tr>
<tr><td>Off</td><td>不显示变量的类型(缺省)</td></tr>
<tr><td rowspan="3">语言</td><td colspan="2">选择显示/不显示指定的变量类型</td></tr>
<tr><td>C</td><td>显示类C的数据(缺省).</td></tr>
<tr><td>ASM</td><td>不能选择</td></tr>
</table>

(13)　功能按钮

OK	保存设置，关闭对话框
Cancel取消	放弃修改并关闭对话框
Restore恢复	恢复对话框打开之前的设置
Help帮助	显示对话框在线帮助文件

工程文件保存对话框

这个对话框用来保存当前的调试环境到工程文件 (参考"5.15.1 调试环境(工程文件)".)
在此对话框中可以保存文件或者覆盖已有文件。

图 6-14：工程文件保存对话框

- 打开
- 每个区域的解释

打开

选择 [File] 菜单中[Project]的 [Save As...]选项.

(为了保存和先前保存或打开文件具有相同名称的工程文件，选择[File]菜单中[Project] 的[Save]选项.)

每个区域解释

(1) 保存文件设置区域

Save in:保存	这个区域用来指定文件的名称。可以直接输入文件名，也可以从此区域上部列表中选择 最多可以指定257字符的字符串作为文件的扩展名
File name:文件名	
Save as type:保存类型	这个区域用来指定保存的工程文件的扩展名 (*.prj)。 如果省略扩展名， "*.prj" 被设为缺省的扩展名。

(2) 功能按钮

Save保存	保存调试环境到指定的文件，文件保存后，关闭对话框
Cancel取消	不保存文件并关闭对话框
Help帮助	显示对话框在线帮助文件

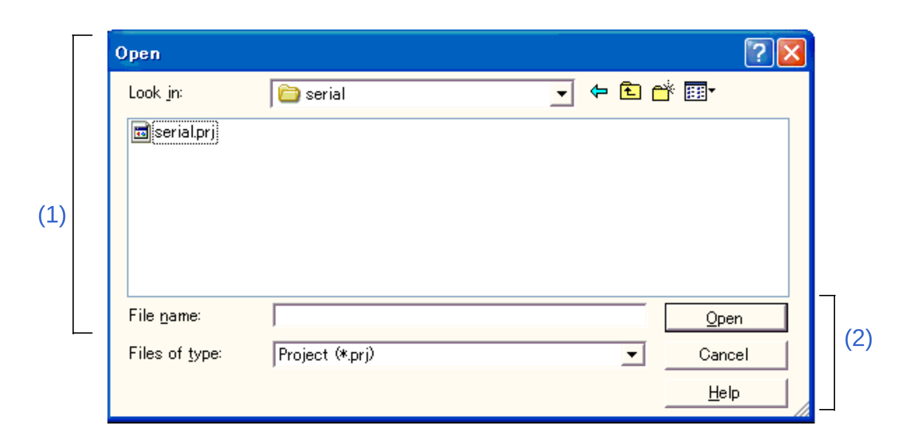
工程文件加载对话框

这个对话框用来恢复调试环境到工程文件中保存的设置。 (参考"5.15.1 调试环境 (工程文件)".)

如果工程文件加载后存在一个激活的源文件窗口，它将显示在最上端。

注意事项: 随着ID850QB的启动，如果指定的工程文件和目标设备启动时曾经使用的工程文件不一致，启动时指定的目标设备将被使用

图 6-15: 工程文件加载对话框



- 打开
- 每个区域解释

打开

单击**Proj** 按钮，或选择[File]菜单中[Project]的[Open...]选项.

每个区域的解释

(1) Load file setting area加载文件设置区域

<table>
<tr><td>Look in:</td><td rowspan="2">这个区域用来指定加载文件的名称。可以直接从键盘输入文件名，也可以从列表中选择。
最多可以指定257字符的字符串作为文件的扩展名。</td></tr>
<tr><td>File name:文件名</td></tr>
<tr><td>Files of type:文件类型</td><td>这个区域用来指定加载文件的扩展名(*.prj)</td></tr>
</table>

(2) Function buttons功能健

Open打开	加载选择的文件，文件加载后关闭对话框。
Cancel取消	不执行任何操作并关闭对话框。
Help帮助	显示对话框在线帮助文件

下载对话框

这个对话框用来选择下载文件的名称和格式，还可以选择在线仿真器和目标系统的下载存储器内容 (参考"5.2 下载功能，上载功能".)

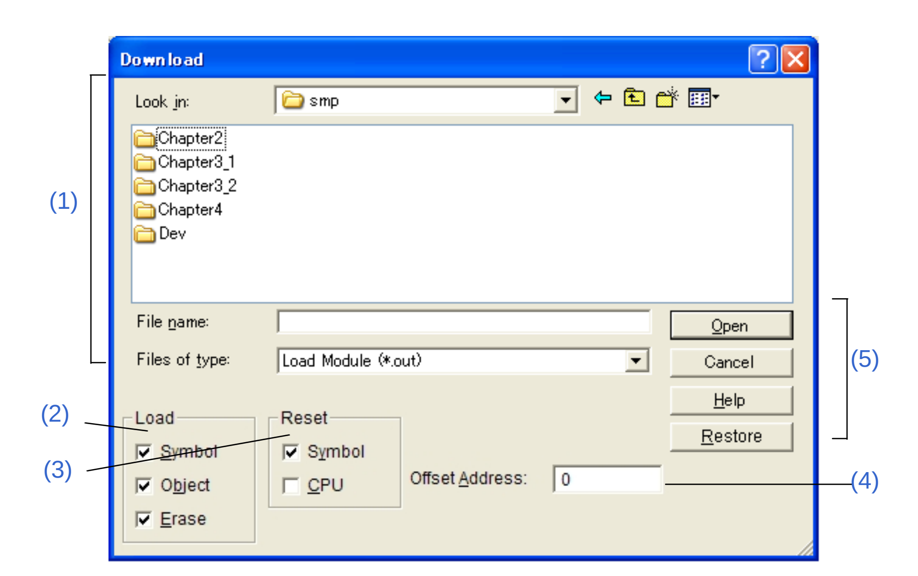
如果已经下载了一个加载模块文件，搜索相应的源文件，并且自动打开源文件窗口。

注意事项: 如果有超过一个的加载模块文件被加载，源程序调试不能执行

备注1: 内部闪存可以被写入，加载模块文件可以被下载 (参考"5.7.4 闪存写入功能 [MINICUBE] [MINICUBE2]"). **[MINICUBE] [MINICUBE2]**

备注2: 当执行下载时，出现下面的对话框，下载可以随时取消。

图 6-16：加载对话框



- 打开
- 每个区域的解释

打开

直接单击Load 按钮，或者选择[File] 菜单的 [Download...]选项。

每个区域的解释

(1)　加载文件设置区域

Look in:	这个区域用来指定加载文件的名称。可以直接从键盘输入文件名，也可以从列表中选择。 最多可以指定257字符的字符串作为文件的扩展名。
File name:文件名	
Files of type:文件类型	这个区域用来指定加载文件的扩展名(*.prj)　(参考"表5-3　可以下载的文件类型".) 这些是缺省的扩展名，其他扩展名也可使用。 在调试选项对话框中可以指定显示的加载模块的扩展名。

备注:　对话框中可以指定两个或多个文件。当指定两个或多个文件时，通过双引号隔开。也可以通过按住Shift或Ctrl键单击鼠标来指定文件。最多可以下载20个加载模块文件。

(2)　加载

设置加载条件。
只有当文件指定为加载模块格式时，设置才是合法的。

Symbol符号	指定是否读取符号信息注
Object目标	指定是否读取目标信息(缺省方式为选择) 。 (当十六进制文件被加载，即使没有选择这个按钮，也会读取目标信息)
Erase删除	不能选择**[IECUBE]** 指定在下载之前是否清除内部闪存的内容 **[MINICUBE]**

注:　当一个程序包含两个或多个需要调试的加载模块文件，并且某些模块的符号信息不必读取的情况下，存储器内容可以在不读入符号信息时被保存。

(3)　复位

设置复位条件。只有文件格式被指定为加载模块文件的情况下，设置才是有效的。

Symbol符号	指定是否复位符号信息注
CPU	指定是否复位CPU(缺省是不选择)

注:　当调试包含两个或多个加载模块文件时，加载完一个文件后不要复位符号信息。当下载两个或多个加载模块文件时，确保文件地址不要重叠。

(4)　偏移地址：

这个区域用来指定加载程序后地址的偏移量（如二进制数据，指定起始地址）。 也可用一个符号或表达式指定地址（参考"表5-6：指定符号"。）缺省的输入方式是十六进制数值。

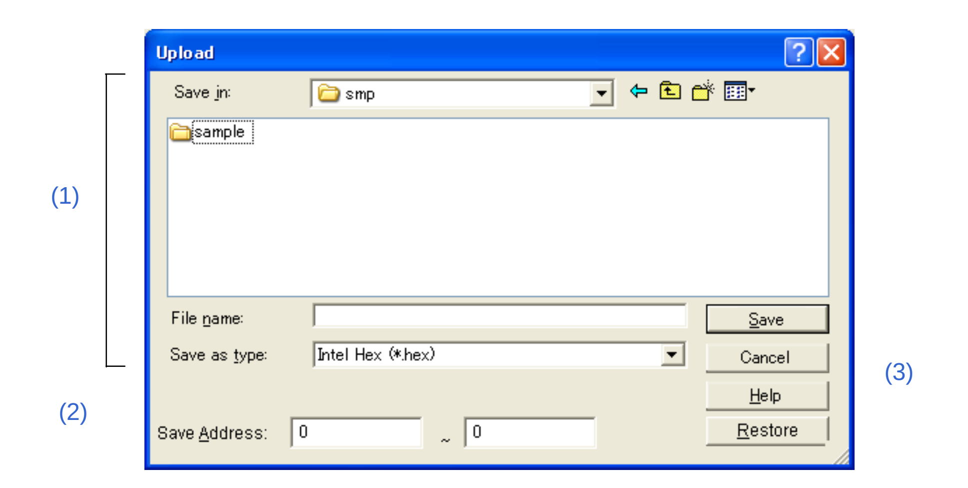
(5)　功能按钮

Open打开	加载选择的文件，当文件加载后，对话框关闭
Cancel取消	不调入文件并关闭对话框
Help帮助	显示对话框在线帮助文件
Restore恢复	恢复输入数据到原始状态

上载对话框

这个对话框用来设置保存文件的名称和格式，这个文件用来保存设置存储器内容 (参考"5.2 下载功能，上载功能".)

图 6-17：上载对话框



- 打开
- 每个区域解释

打开

选择[File] 菜单中[Upload...]选项.

每个区域域解释

(1)　上载文件设置区域

Save in:保存 File name:文件名	这个区域用来指定文件名称。可以直接输入一个文件名称，也可以这个区域的上部通过列表选择。 最多可以输入257字符作为文件扩展名。
Save as type:文件类型	这个区域用来指定保存文件的扩展名。保存的数据格式由扩展名确定。 (参考"表5-4：可以上载文件的类型".)

备注:　　其他的扩展名也可以使用。

(2)　保存地址

这个区域用来指定保存的地址范围。

当覆盖率数据(*.cvb)被选中时，所有的区域都被保存 (这个区域不可以设置)

地址也可以用符号或表达式来指定。(参考"Table 5-6 指定符号".) 输入的缺省的基数是十六进制。

(3)　功能按钮

Save保存	根据设置保存文件
Cancel取消	不作修改并关闭对话框
Help帮助	显示对话框在线帮助文件
Restore恢复	在对话框打开之间恢复状态

加载模块列表对话框

这个对话框用来显示从下载对话框中下载的文件、文件路径和文件格式列表 (参考"5.2 下载和上载功能".)

这些列表文件 (不包括覆盖率数据文件)被保存在工程文件中。在工程文件打开后它们被下载。通过使用<Download...>按钮，可以打开下载对话框并下载文件。

图 **6-18**：加载模块列表对话框

- 打开
- 每个区域解释

打开

在[File]菜单中选择 [Load Module...]按钮.

每个区域解释

(1)　加载模块文件显示区域

区域			说明
Module list模块列表			这个区域显示已经下载的文件名称
	No.		已经读取的加载模块文件的指示序列号。
	Format		显示的文件格式
		BIN	二进制文件
		HEX	十六进制文件
		COV	覆盖率数据文件
		ELF/CA850	加载模块文件(ELF/CA850)
		ELF/DWARF2	加载模块文件（ELF/GHS扩展DWARF2)
		ELF	ELF加载模块文件（没有符号信息的ELF)
		Unknown	未知
	File name文件名		如果选择"Add Path" ，将按全路径显示文件名称，否则只显示文件名称
Add Path			选择文件名字与路径一起显示。
<Download...>			打开下载对话框。 可以下载一个新的加载模块文件。 当下载对话框被关闭的时，新文件的文件名字将会添加到文件名字显示区域。

注意事项:　　如果在复位调试器对话框中复位符号信息，或者在下载对话框中复位符号信息，以前下载的文件名将被清除

(2)　功能按钮

按钮	说明
Close关闭	关闭对话框
Help帮助	显示对话框在线帮助文件

源文件窗口

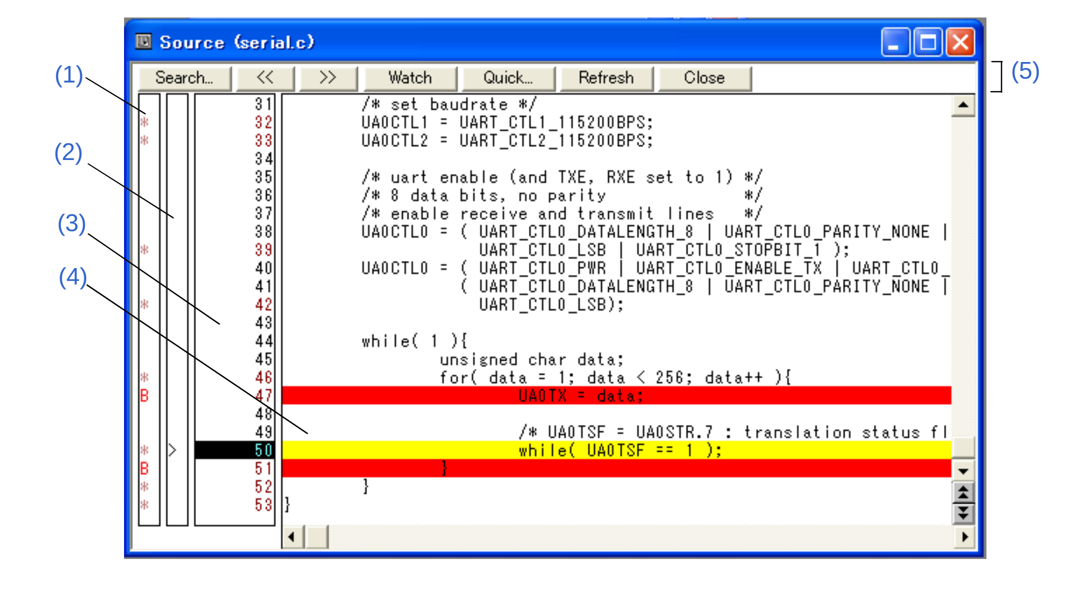
这个窗口用来显示源文件或者文本文件。 (参考"5.3 源文件显示和反汇编显示功能".)，另外还有断点设置，覆盖率测量执行位置显示和混合显示功能(源文件窗口)。其它的操作可以通过这个窗口中上下文菜单、功能按钮来实现。而且这个窗口具有静态和激活两个状态。当窗口是激活状态时，它具有关联窗口的跟踪结果[IECUBE]. 而且这个窗口中通过拖&放功能选择的项目可以在其它窗口中使用 (参考"5.16 每个窗口的通用功能")。

注意事项1: 如果程序代码是在一个包含文件中描述，并包含在多个文件中，行号和地址不是一一对应。 在这样的包含文件中，指示行号和地址对应关系的功能可能给出错误的结果。

注意事项2: 如果下载加载文件模块后，包含"main"函数的源文件没有在源路径中找到，或者单步执行期间不能找到源文件，ID850QB为选择源文件打开一个对话框，并提示用户为对话框中显示的源文件选择路径。如果单击<Cancel>按钮，显示的文件名称将被记忆，因此直到ID850QB终止都不会询问源文件名称。

备注: 最多可以显示65535行C或者汇编语言源程序。如果源程序超过65535行，进行程序分割。

图 6-19：源文件窗口



- 打开
- 每个区域的解释
- [View] 菜单(源文件窗口专用功能)
- 上下文菜单

打开

单击**Src** 按钮或选择[Browse]菜单的 [Source Text]选项.

(当下载文件模块被加载后，如果存在相应的源文件，系统将自动打开这个窗口)

每个区域解释

(1)　点标记区域

这个区域用作事件设置状态(事件标记)和程序代码(*)显示，以及断点设置。

通过在程序代码处单击鼠标可以设置断点和删除断点 (如果这一行没有显示"*"，断点设置在显示"*"的上一行或者下一行。)

只有当加载模块文件下载的符号信息被读入时，程序代码才能显示。

如果在相应的行已经设置事件，下面表中的一个标记将显示。根据断点类型和状态不同，"B"标记的颜色也不同。 (当这个区域设置一个断点时，同时它也能被使能)

表 6-11：事件设置状态 (事件标记)

标记	含义
B (blue)	已经设置软件断点。
B (red)	已经设置有效的硬件断点 (执行后)。
B (green)	已经设置有效的硬件断点(执行前)。 **Note**: 执行前的断点被设置为高优先级。
B (black)	已经设置无效的硬件断点。 在事件管理器和断点对话框中可以使硬件断点有效。
E	事件条件已经设置
L	事件关联条件已经设置
T	跟踪事件已经设置
Ti	定时器事件已经设置
A	多重事件已经设置

备注:

如果指定地址范围为事件的地址条件，显示范围的低地址。地址条件的屏蔽不能反映出来。

(2) 当前PC标记区域

指示当前PC值的">"标记，在此区域中显示。

用鼠标点击此标记，显示一个弹出窗口，显示PC寄存器值。

双击当前PC标记区域，程序可以执行到一个指定行。(参考" [Come Here]"。)

(3) 行号/地址显示区域

此区域显示源文件或者文本文件的行号。

红色指示相应的程序代码行存在，黑色指示相应的程序代码行不存在。在混合显示模式(源文件窗口)，反汇编显示地址显示为灰色。

(参考"5.11.3 显示覆盖测量执行的位置".)另外，基于代码覆盖率测量信息，执行地址高亮显示。

(4) 源文本显示区域

此区域显示源文件或者文本文件的行号。以下行重点显示。

当前PC行(黄色)[注]	黄色指示当前PC行(或者反汇编行) ('>')。如果断点事件发生时，在PC位置处如果没有行号信息，立即跳转到汇编窗口执行。在混合显示模式(源文件窗口)，只有反汇编显示行颜色发生改变(源文件行以正常方式显示)。
段点设置行(红色)	红色指示有效断点设置的行。在混合显示模式(源文件窗口)，只有反汇编显示行颜色发生改变(源文件行以正常方式显示)。

而且，这一个区域也提供下列的行功能 (程序代码的起始地址) 和光标已经被放置的地址。

- [Come Here], [Start From Here] (参考"表5-9：执行的类型".)
- 拖动和放置功能
- 上下文菜单

注意事项: 如果一个程序代码在源行上不存在, 程序代码存在的上面的行或者下面的行将被操作。
这些功能不能够在下列的情形运行。　对应的菜单将会被暗淡而且不能够选择。
- 如果源文件之外的一个文件被显示
- 当用户程序正在运行的时候

(5) 功能按钮

Search... 搜索	打开源搜索对话框而且搜索源文本的一个字符串。 如果在源文本显示区域选择一个字符串，将打开源搜索对话框搜索字符串。 如果没有选择字符串，源搜索对话框在无指定搜索内容的情况下打开。 在源搜索对话框中指定一个搜索方法。在源窗口高亮显示搜索结果。 通过选择[View] 菜单的 [Search...]项执行同样的操作。
<<	前向搜索(屏幕向前)满足源搜索对话框中设置的搜索条件的文本。 搜索从在光标位置的地址处开始。 在搜索期间这个按钮被显示为<Stop> 按钮。
>>	后向搜索(屏幕向后)满足源搜索对话框中设置的搜索条件的文本。 搜索从光标位置的地址处开始。 在搜索期间这个按钮被显示为<Stop> 按钮。
Stop (during a search)停止(搜索期间)	停止搜索
Watch观察	增加源文本显示区域的选择变量到观察窗口。 如果没有打开观察窗口，则打开观察窗口。 如果在源文本显示区域没有选择文本，仅仅打开观察窗口。 选择[View]菜单的 [View Watch]选项执行同样操作。
Quick... 快速	在快速观察对话框中临时显示诸如变量，选择的原文本等内容。 如果在源文本显示区域没有选择文本，仅仅打开快速观察窗口。 选择[View]菜单的[Quick Watch...] 选项执行同样操作。
Refresh更新	用最近的数据更新窗口的内容。
Close关闭	关闭窗口

[View] 菜单(源文件窗口专用功能)

当源文件窗口处于激活状态，下列项目自动加入[View]菜单中。

Create Break Event创建断点事件	设定一个断点事件，如果选择的变量被存取，此断点事件发生。
Break when Access to this Variable访问此变量时产生断点	设定一个断点事件。如果选择的变量被读写，此断点事件发生。
Break when Write to this Variable写入此变量时产生断点	设定一个断点事件。如果选择的变量被写入，此断点事件发生。
Break when Read from this Variable此变量读出时产生断点	如果选择的变量被读存取，设定一个断点事件。如果选择的变量被读出，此断点事件发生。
Clear清除	删除选择变量相应的断点事件。
Event Information事件信息	显示光标位置的行的事件信息或选择的变量名称。 如果设置了事件，打开事件对话框。
Mix混合	打开/关闭混合显示模式 (源文件窗口).

上下文菜单

Move... 移动	移动显示位置 打开源文本移动对话框。
Mix混合	关闭/打开混合显示模式 (源文件窗口)。
Add Watch... 添加观察	添加指定的数据到观察窗口。 打开添加观察对话框。
Symbol... 符号	显示指定变量或者函数的地址，或者指定符号的数值。 打开符号地址对话框。
Break when Access to this Variable访问此变量时产生断	设定一个断点事件。如果选择的变量被读写，此事件发生。
Break when Write to this Variable写入此变量时产生断	设定一个断点事件。如果选择的变量被写入，此事件发生。
Break when Read from this Variable读出此变量时产生断	设定一个断点事件。如果选择的变量被读出，此事件发生。
Clear清除	删除选择变量相应的断点事件。
Event Information事件信息	显示光标位置的行的事件信息和选择的变量名称。 如果设置了事件，打开事件对话框
Come Here运行到	从当前的PC值开始执行程序到光标位置 (参考"表5-7 断点类型".)
Change PC改变PC	将当前光标处的地址设置为PC值。
Break Point断点	删除光标位置处的硬件断点 **Note:** 执行(B)前的断点被设为高优先级
Software Break Point软件断点	在光标处设置和删除软件断点

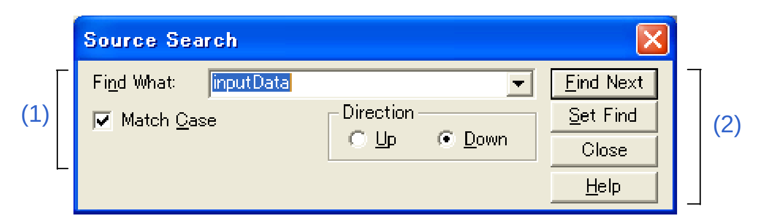
Assemble汇编	从以光标位置处的数据值指定的跳转目的地址开始反汇编和显示。(参考"5.16.2跳转功能") 打开汇编窗口 如果汇编窗口打开，这个窗口在前面显示(因此可以对其操作).
Memory内存	从以光标位置处的数据值指定的跳转目的地址开始显示存储器内容。(参考"5.16.2跳转功能"。) 打开存储器窗口。 如果一个存储器窗口被打开，该窗口显示在最前面(以便操作)。

源搜索对话框

这个对话框用来在源文件窗口中搜索文件的内容 (参考"5.3.1 源文件显示".)

通过设置每一项，单击<Find Next> 按钮可开始搜索。通过单击<Set Find> 按钮，使用源文件窗口中的方向按钮 ("<<" 和 ">>")进行搜索。

图 6-20：源搜索对话框



- 打开
- 每个区域解释

打开

当源文件窗口是当前窗口时，选择 [View] 菜单中的[Search...]选项，或在同一窗口中单击 <Search...> 按钮。

每个区域解释

(1) 搜索条件指定区域

Find What:搜索	这个区域用来指定搜索的数据 (最多256字符。) 缺省方式下，显示调用此对话框的显示窗口中选择的字符串。 如必需时，可修改显示的字符串内容。最多记录16次输入历史记录。	
Match Case	这应该被选择区别大写字母的和小写字母。	
Direction方向	这个区域用来指定搜索的方向	
	Up	前向搜索。 搜索数据从光标的目前位置向前 (屏幕中向上)
	Down	后向搜索。 搜索数据从光标的目前位置向后 (屏幕中向下)(缺省)

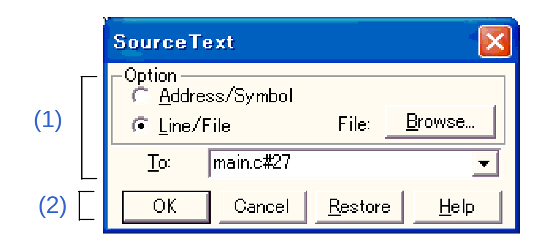
(2) 功能按钮

Find Next搜索下一个	搜索符合一个给定的条件指定数据。如果发现指定的字符串是一个搜寻的结果,将其高亮显示。 继续搜索，再次单击这个按钮.
Set Find搜索设定	指定的条件设定为搜索条件，关闭对话框
Stop (during searching) 停止(搜索期间)	停止搜索
Close关闭	关闭对话框 (搜索期间，这个按钮被<Stop>按钮代替)
Help帮助	显示此对话框在线帮助文件

源文本移动对话框

这个对话框用来指定源文件窗口中显示的文件和显示文件的起始位置(参考"5.3.1 源显示".)

图 6-21：源文本移动对话框



- 打开
- 每个区域解释

打开

当源文件窗口是当前窗口时，选择[View]菜单中的[Move...]选项.

每个区域解释

(1)　源文件设置区域

<table>
<tr><td rowspan="3">Option操作</td><td colspan="2">指定起始位置的情况下选择输入模式。</td></tr>
<tr><td>Address/Symbol地址/符号</td><td>应该被选中，用地址(或者符号)指定。</td></tr>
<tr><td>Line/File行/文件</td><td>应该被选中，用行号(或者文件名)指定。使用<Browse...>按钮搜索文件名。</td></tr>
</table>

<table>
<tr><td rowspan="3">To:</td><td colspan="2">指定要显示的文件名字或者地址。 最多可记录16个
输入历史记录。</td></tr>
<tr><td>When "Address/ Symbol" is selected当"地址/符号"被选中</td><td>指定显示的起始地址。
输入数据的缺省基数十六进制。可使用符号或者表达式指定地址 (参考"表5-6 指定符号"。)
单击 <OK> 按钮显示源本文，以便观察指定地址值相应的行。</td></tr>
<tr><td>When "Line/File" is selected当"地址/符号"被选中</td><td>指定显示开始的行号(或者文件名)。
通过 [[path name路径名字] file name文件名字]# line number行号来指定行号。
缺省方式下输入数值的基数是十进制数。
仅使用文件名字，或使用绝对的路径和相对路径来指定文件名字。 如果只是文件名字或相对的路径被指定，搜寻在调试选项对话框内被指定的文件。
点击<OK>按钮，指定的行号被指定为要显示的第一行。 当文件名字被省略的时候，现在显示的文件从指定的行被显示。 如果行号被省略,文件从第一行开始显示。</td></tr>
</table>

(2)　功能按钮

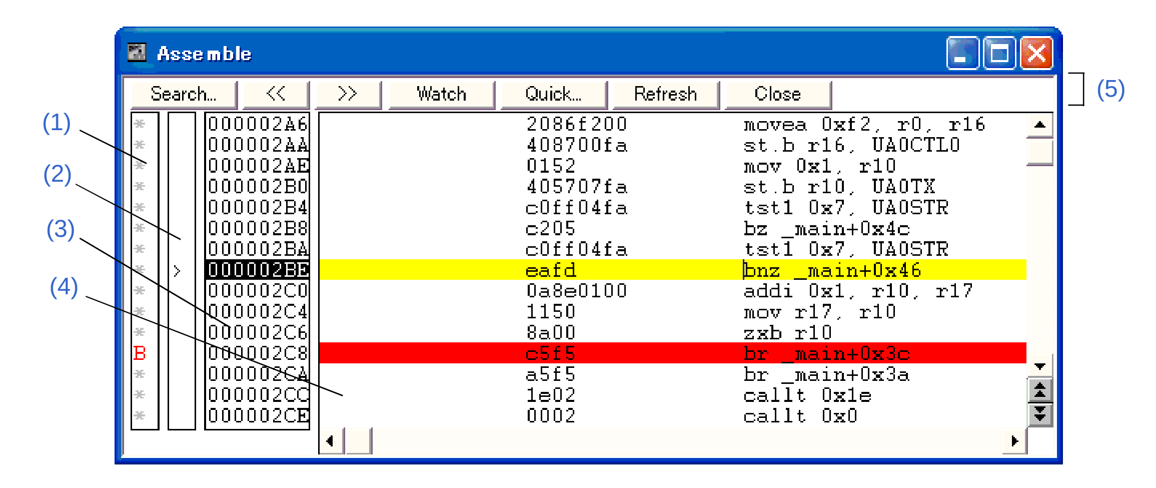
OK	从指定的位置显示源文件文本
Cancel取消	关闭对话框
Restore恢复	恢复原始状态
Help帮助	显示对话框在线帮助文件

汇编窗口

这个窗口用来反汇编和显示程序，也可以用来执行在线汇编 (参考 "5.3 源文件显示和反汇编显示功能".)　在存储器窗口也会显示在线汇编的结果。

除了断点设定和覆盖率测量执行位置的显示之外，还可以在这一窗口中执行一些其他的操作，如使用上下文菜单，功能按钮，等。而且，对于此窗口有激活状态和静态状态二种状态。当窗口处于激活状态时, 具有和窗口相关的跟踪结果[IECUBE]。 而且，通过拖&放功能选择的项目在另一个窗口也可使用(参考"5.16 窗口的公共功能".)

图 6-22：汇编窗口



- 打开
- 每个区域解释
- [View] 菜单(汇编窗口专用功能)
- 上下文菜单
- 相关的操作

打开

单击**Asm按钮**或选择[Browse]菜单中的[Assemble]选项.

每个区域解释

(1)　点标记功能

这个区域用来实现设置事件设置状态(事件标记)和断点设置。

(2)　当前PC标记区域

用来指示当前PC值(PC寄存器值)的标记">"在此区域显示。

双击当前PC标记区域，程序将执行到指定的行 (参考"[执行到这里]".)

(3)　地址指定区域

这个区域用来显示反汇编的起始地址

此外，基于代码覆盖率测量信息的已执行地址将以高亮的方式显示。 (参考"5.11.3 对已执行覆盖测量部位的显示".)

备注:　　结束地址是不显示的，结束地址为0xFFFFFFFE.

(4)　反汇编显示区域

这个区域用来显示地址的标号和代码，及反汇编助记符。

在助记符区可以实现在线汇编。

下面是需要强调的内容。

当前PC行 (黄色)	黄色指定当前PC行 ('>').
断点设置行(红色)	红色指定设置有有效断点的行

这个区域也提供下列功能：

- [执行到这里]，[从这里开始执行] (参考"表5-9：执行的类型".)
- 拖&放功能
- 上下文菜单

(5)　功能按钮

Search...搜索	打开汇编搜索对话框，搜索助记符字符串。 在汇编搜索对话框中指定搜索方式。搜索结果将在汇编窗口中以高亮的方式显示。 选择[View]菜单中 [Search...]选项执行同样的操作。
<<	根据汇编搜索对话框中设定的搜索条件从当前光标处向前搜索(屏幕向前) 在搜索期间这个按钮显示为<Stop>。
>>	根据汇编搜索对话框中设定的搜索条件从当前光标处向后搜索(屏幕向后) 在搜索期间这个按钮显示为<Stop>。
Stop(during a search) 停止(搜索期间)	停止搜索
Watch观察窗口	添加在(4)反汇编显示区域选择的符号到观察窗口。如果观察窗口没有打开，将打开观察窗口。 如果"反汇编显示区域"没有指定内容，将只打开快速观察对话框。 选择[View]菜单中[View Watch...]选项将执行同样的操作。.
Quick...快速观察	临时显示快速观察对话框中(4)反汇编显示区域中选中的内容，如符号。 打开快速观察对话框。如果"反汇编显示区域"没有指定内容，将只打开快速观察对话框。 选择[View]菜单中[Quick Watch...]选项将执行同样的操作。.
Refresh刷新	以最新数据刷新当前窗口内容。
Close关闭	关闭窗口。

[View] 菜单(汇编窗口专用功能)

当汇编窗口是激活状态时，下列内容将加入到 [View] 菜单中。

事件信息	显示光标所在地址处的事件信息。 如果设置了事件，打开事件对话框。

上下文菜单

菜单项对于选择的行或者项有效，而不是鼠标所点击的位置。

(在主菜单中选择同样的名称选项执行同样的操作).

Move...移动	移动显示位置 打开地址移动对话框
Add Watch...添加观察窗口	添加指定的数据到观察窗口。打开添加观察对话框。
Symbol...符号	显示指定变量或函数的地址，或者指定符号的值。 打开符号转地址对话框。

Come Here运行到	从当前的PC开始运行程序到光标位置（参考"表5-7：断点类型"。）
Change PC修改PC	将光标位置处的地址设置为PC。
Break Point断点	设置或删除光标处的硬件断点。 **Note:** 执行前的断点(B)具有高优先级。
Software Break Point软件断点	设置或删除光标处的软件断点。
Source Text源文件	显示相应的源文件和源代码行，将光标处的数据值设定为跳转目的地址（参考"5.16.2 跳跃功能"。） 但是，如果跳转的目的地址没有行信息，将不能执行跳转操作。 打开源文件窗口。 如果打开了一个激活的源文件窗口，这个窗口在最前面显示(因此可以对其进行操作).。
Memory存储器	从跳转目的地址开始显示存储器内容，跳转目的地址用当前光标的位置来设定(参考"5.16.2 跳转功能"。) 打开存储器窗口。 如果打开了一个激活的存储器窗口，这个窗口在最前面显示(因此可以对其进行操作).。

相关的操作

(1) 在线汇编

要改变反汇编的内容，移动光标到助记符区域(按下插入键切换覆盖和插入模式)。

助记符区域的反汇编内容被修改后，如果试图移动光标到另一行，修改后的内容将被检查。如果新内容是非法的，修改过的行的代码数据将被标记"*"。

通过按下回车键，助记符区域中修改过的内容将写入存储区。按下回车键，将检查修改的内容。即使只有一行是非法的，新内容也不能写入存储区。通过按下ESC键，放弃这些内容。

如果修改的内容正确并且按下回车键，这些内容将写入存储区，然后光标会移动到助记符区域的下一行，以便对下一行的数据进行修改。

注意事项: 如果修改导致新指令字节个数比以前的指令字节个数减少，将插入尽可能多的'nop'指令。如果新指令字节比以前的指令字节个数多，下一条指令被覆盖。这种情况下，同样需要插入尽可能多的'nop'指令。对源文件中跨行指令同样适用。

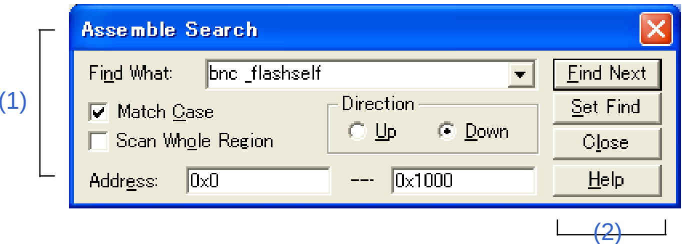
汇编搜索对话框

这个对话框用来对汇编窗口中的内容进行搜索 。(参考"5.3.2 反汇编显示"。)

输入字符串和反汇编字符串中的连续的字符串将作为一个空字符进行比较。

通过设置相应地选项，单击<Find Next>按钮，将开始执行搜索。在汇编窗口中可以使用<Set Find> 按钮，方向按钮 ("<<"和">>") 进行搜索。

图 6-23：汇编搜索对话框



- 打开
- 每个区域的解释

打开

当汇编窗口是当前窗口时，选择[View]菜单中的[Search...]选项，或者在同一窗口中单击<Search...>按钮。

每个区域的解释

(1)　搜索条件指定区域

Find What: 搜索内容	这个区域用来指定搜索的数据(最多256字符。) 在缺省方式下，显示调用此对话框的窗口中选择的字符串。 必要时，可以修改显示的字符串，最多记录16个输入的历史记录	
Match Case大小写匹配	选择此项以区分大小写。	
Scan Whole Region搜索整个区域	选择此项搜索整个指定范围。	
Direction方向	这个区域用来指定搜索的方向	
	向上	前向搜索。从当前光标处向前搜索数据 (屏幕向前)
	向下	后向搜索。从当前光标处向前搜索数据 (屏幕向后)

Address:地址	这个区域用来指定搜索的地址 输入数据的缺省数制为十六进制。也可以用一个符号或者表达式指定地址 (参考"表 5-6 指定符号".)

(2)　功能按钮

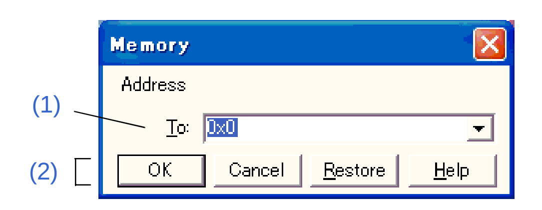
Find Next搜索下一个	根据给定的条件搜索指定的数据。 如果指定的字符串被搜索到，结果以高亮的方式显示。 要继续搜索，单击这个按钮
Set Find设置搜索	设置指定条件为搜索条件，然后关闭对话框。
Stop (searching)停止(搜索)	停止搜索
Close关闭	关闭对话框(搜索期间，这个按钮被 <Stop>代替)
Help帮助	显示对话框在线帮助文件

地址移动对话框

这个对话框用来指定如下显示的起始地址.

- 存储器窗口
- 汇编窗口
- IOR窗口

图 6-24：地址移动对话框（例：当存储器窗口打开时)



- 打开
- 每个区域的解释

打开

当目标窗口是当前窗口时，选择[View] 菜单中[Move...]选项.

每个区域的解释

(1) 地址指定区域

Address地址	这个区域用来指定显示的起始地址	
	To:	缺省条件下，显示调用此对话框的窗口中选中的字符串或当前PC值，等等。必要时显示的字符串可以被修改。 输入数据的缺省数制是十六进制。也可以用符号或者表达式指定地址(参考 "表5-6 指定符号".) 最多可以记录16个输入历史记录。

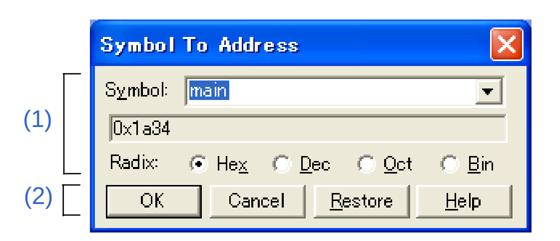
(2) 功能按钮

OK确认	在此地址显示相应的窗口。
Cancel取消	关闭对话框。
Restore恢复	恢复对话框打开之前的设置状态。
Help帮助	显示对话框在线帮助文件。

符号转地址对话框

这个对话框用来显示指定变量地址或者指定符号的值(参考"5.3.4 转换符号(符号到地址)".)

6-25：符号到地址转换对话框



- 打开
- 每个区域的解释

打开

选择[View]菜单中的 [Symbol...]选项.

每个区域解释

(1) 符号转换区域

Symbol:符号	这个区域用来指定要转换的变量、函数名称、符号名称或行号 (参考"表5-6 指定符号".) 要修改这个区域的内容，单击 <OK> 按钮。在下面的区域中显示转换结果。 输入数据的缺省数制是十进制。最多可以记录16个输入的历史记录。
Conversion result display area转换结果显示区域	如果指定了位(bit)符号，它们被转换为地址.比特(Address.bit)格式。但是，不能指定包括位符号的表达式。显示的内容有变量、函数地址，符号的值，行号的地址或在"Symbol："中指定的表达式的值。也可以显示周边I/O寄存器的地址、寄存器的内容和PSW标志的值。
Radix:数制	这个区域用来指定要显示的转换数据的数制 Hex：十六进制数(缺省) Dec：十进制数 Oct：八进制数 Bin：二进制数

(2)　功能按钮

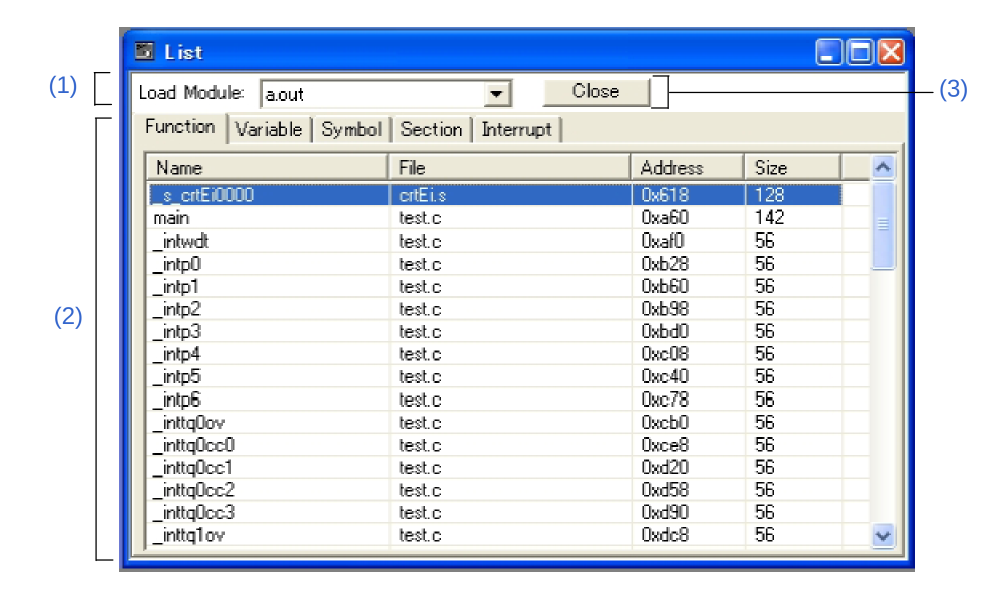
OK确认	如果"Symbol:"中的内容被修改，对其进行转换。转换结束后，如果"Symbol:"中的内容没有被修改，关闭对话框。
Cancel取消	关闭此对话框。
Restore恢复	恢复输入数据到原始状态。 如果已单击<OK>按钮，数据被恢复到刚刚单击<OK>按钮后的状态。
Help帮助	显示对话框在线帮助文件

列表窗口

这个窗口用来实现函数、变量、符号、段和中断请求的列表。

显示的数据可以不依赖于制表符而存储为CSV格式 (参考"5.15.2 窗口显示信息(查看文件)".)

图 6-26：列表窗口



- 打开
- 每个区域解释
- 上下文菜单

打开

选择[Browse]菜单中的[List]选项.

每个区域的解释

(1)　加载模块:

这个区域用来选择下载的加载模块文件。当没有下载加载模块文件时，这个区域是空的。

(2)　列表查看区域

这里通过每个制表符列表显示诸如名称、大小、函数地址、变量、符号、节和中断请求。下载加载模块文件后将自动更新显示的内容。

如果没有下载加载模块文件，这个区域是空的。

(a) 当选择[Function]制表符时

Name名称	函数名称 (在汇编源文件中以文件为单位显示函数)
File文件	定义函数所在文件的名称
Address地址	函数起始地址
Size大小	函数大小（单位：字节)

(b)　当选择[Variable]制表符时

Name名称	变量名称
File文件	定义变量所在文件的名称
Address地址	变量的起始地址
Size大小	变量大小（单位: 字节)

(c)　当选择[Symbol]制表符时

Name名称	符号名称
Address地址	符号地址

(d)　当选择[Section]制表符时

Name名称	节名称
Type类型	节类型(代码，数据)
Address地址	节的起始地址
Size大小	节的大小(单位：字节)

(e)　当选择[Interrupt]制表符时

Name名称	中断请求名称
Type类型	中断类型 (非屏蔽，可屏蔽，软件，安全ID，闪存屏蔽操作)
Status状态	程序中的使用状态 (使用，未使用) ----：未知
Address地址	中断句柄的起始地址
Size大小	中断句柄的大小 (单位：字节) 除"use"外其它状态的最大值

可以通过简单的双击跳转源地址跳到上面任一窗口中。

使用选中的行的起始地址值作为一个跳转指针，从当前制表符跳转到源文件窗口，汇编窗口或者存储器窗口。

跳转功能的执行方法：选中一个跳转源文件行，然后在[Jump]菜单中选择[Source Text/Assemble/Memory]。

备注:　　通过在每一列中单击标题(在标签上)对显示项目进行排序(每次单击标题，升序/降序次序切换一次)

(3)　功能按钮

Close关闭	关闭此窗口

上下文菜单

Source Text源文本	使用光标处的数值作为跳转目的地址，显示相应的源文本和源文件行。(参考"5.16.2 跳转功能")然而，如果在跳转目的地址处不存在行信息，不能跳转。 打开源文件窗口。如果打开了一个激活的源文件窗口，该窗口在最前面显示(因此可以操作)。
Assemble汇编	从光标位置处的数据值指定的跳转地址处开始反汇编和显示。(参考"5.16.2跳转功能") 打开汇编窗口。如果打开了一个激活的汇编窗口，该窗口在最前面显示(因此可以操作)。
Memory存储器	从光标位置处的数据值指定的跳转地址处开始反汇编和显示。(参考"5.16.2跳转功能") 打开汇编窗口。如果打开了一个激活的汇编窗口，该窗口在最前面显示(因此可以操作)。

观察窗口

这个窗口用来显示和修改指定的观察数据 (参考"5.6 观察功能".)

类似存储器窗口，即使在程序执行期间，这个窗口可以实时显示众多的观察数据 (诸如全局变量和公用符号)。

在此窗口更新和重写数据将反映在存储器窗口中。

在源文件窗口或者汇编窗口中单击 <Watch...>按钮可以登录观察数据 (参考"5.6.3 登录和删除观察数据".)

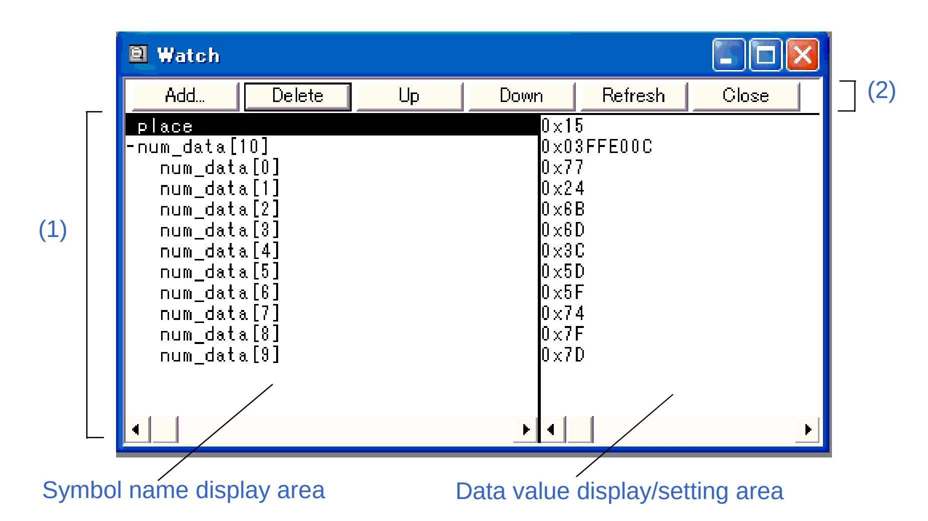
利用这个窗口的上下文菜单可以容易地对一个变量设置断点

备注1:　　如果局部变量和全局变量具有相同的名称，局部变量优先处理。

注意事项:　　如果在用户程序执行期间，RRM区域中的跨边界(256 bytes)区域被访问，高端地址将显示无效值。因而，相应的变量在观察窗口中高亮显示为红色。(这种情况下，存储器窗口中的显示值也同样无效，但是它们没有高亮为红色)。只有在用户程序执行期间，这种情况才发生。在断点时显示正常值。**[IECUBE]**

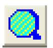
备注2:　　观察窗口中最多可以显示10,000 行。

图 6-27：观察窗口



- 打开
- 每个区域的解释
- [View] 菜单 (观察窗口专用功能)
- 上下文菜单

打开

 单击**Wch** 按钮或选择[Browse] 菜单中[Watch]选项。

每个区域解释

(1) 观察数据显示/改变区域

左边的区域显示符号名称，右边的区域显示数据值。

<table>
<tr><td rowspan="4">Symbol name display area符号名称显示区域</td><td colspan="2">这个区域被用来显示变量名字，符号名字或者类型，以及结构或者联合的标记名称。'+'作为前缀附加到显示数组，指针变量，结构或者联合之上。当双击这些变量时，将展开并显示这些变量(第一个字符由"+"变为"-")。
通过在修改观察对话框中选中并打开要修改的项目然后选择上下文菜单->修改观察窗口，修改登录的观察数据。
如果在观察窗口中为一个变量或者符号设置了访问断点，符号名字显示区域高亮显示为金黄色。</td></tr>
<tr><td>数组</td><td>根据数组变量的类型，通过双击"+"，显示所有的变量元素。</td></tr>
<tr><td>指针变量</td><td>通过双击"+"，显示指针指向的数据。</td></tr>
<tr><td>结构/联合</td><td>根据变量成员类型，通过双击"+"，显示所有的结构/联合成员。
如果在结构或者联合中定义了一个结构或者联合，也显示内部结构或者联合的结构或者联合名字。
内部结构或者联合同样可以使用"+"扩展。</td></tr>
</table>

<table>
<tr><td>Data value display/setting area数据值显示/设置区域</td><td>此区域用来显示和修改观察数据值。当停止运行时，数值被更新。选择[文件] 菜单->[保存为...]，保存数值。获取数据失败时，此区域为空。通过直接输入可以修改数值。要修改的位置显示为红色，当回车键按下后修改的内容将写入目标存储区。以前设置的值通过ESC键取消。
<table>
<tr><th>显示数据</th><th>内容</th></tr>
<tr><td>整数</td><td>十六进制(0xxxxx)
十进制 (xxxx)
八进制 (0xxxx)
二进制 (0bxxxx)</td></tr>
<tr><td>字符型</td><td>"Character"</td></tr>
<tr><td>枚举类型</td><td>成员名称</td></tr>
<tr><td>如果指定了范围</td><td>根据指定的范围显示。</td></tr>
<tr><td>浮点类型</td><td>支持单精度/双精度。输入/显示格式如下：
[+ | -] inf
[+ | -] nan
[+ | -] integer e [+ | -]exponent
[+ | -] integer.fraction[e [+ | -]exponent</td></tr>
<tr><td>"?"</td><td>因为范围内有改变或者化编译而导致的无效数据。</td></tr>
</table>
</td></tr>
</table>

备注1:　如果一个数组具有太多的变量不能展开，系统将弹出一个告警信息。

备注2:　在上下文菜单中，可以改变每个变量的数制。在调试选项对话框中可以修改整数的格式和数组中元素的数目。

(2)　功能按钮

Add...添加	打开添加观察对话框 如果数据已指定，单击添加观察对话框中的<Add...>按钮，将被添加指定数据到观察窗口。
Delete删除	从窗口中删除选择的数据。
Up向上	将选择的行向上移动。
Down向下	将选择的行向下移动。
Refresh刷新	以最新数据更新窗口的显示。
Close关闭	关闭窗口

[View] 菜单(观察窗口专用功能)

当这个窗口是当前窗口时，下列条目被增加到[View] 菜单中。 只有操作选中的条目。

Create Break Event创建断点事件	通过选择下面的项创建断点事件。
Beak when Access to this Variable访问此变量时产生断点	创建选中变量的读写访问断点。
Break when Write to this Variable写入变量时产生断	创建选中变量的写访问断点。
Break when Read from this Variable从变量读出时产生断点	创建选中变量的读访问断点。
Clear清除	删除指定项目相应的断点
Event Information事件信息	显示选择变量的事件信息，如果设置了事件，将打开事件对话框。
Bin二进制	用二进制显示选择的行
Oct八进制	用八进制显示选择的行
Dec十进制	用十进制显示选择的行
Hex十六进制	用十六进制显示选择的行
String字符串	显示选择的行为一个字符串
Proper	显示选择的行中每个变量的缺省值 根据调试选项对话框的设置显示符号(缺省).
Byte字节	以8比特为单位显示选择的行。
Half Word半字	以16比特为单位显示选择的行。
Word字	以32比特为单位显示选择的行。
Adaptive适应	显示选择的行中每个变量的缺省值(缺省)。 对于C语言符号只有这一项是有效的选择。 根据调试选项对话框的设置显示汇编语言中符号。
Up向上	选择的行向上移动一行。
Down向下	选择的行向下移动一行。
Compulsion Read强制读取	强制读取那些为避免数值改变而禁止读取的周边I/O寄存器，或者在添加I/O端口对话框中添加的I/O端口和I/O保护区域的数据。

上下文菜单

菜单项对于选择的行或者项有效，而不是鼠标所点击的位置。

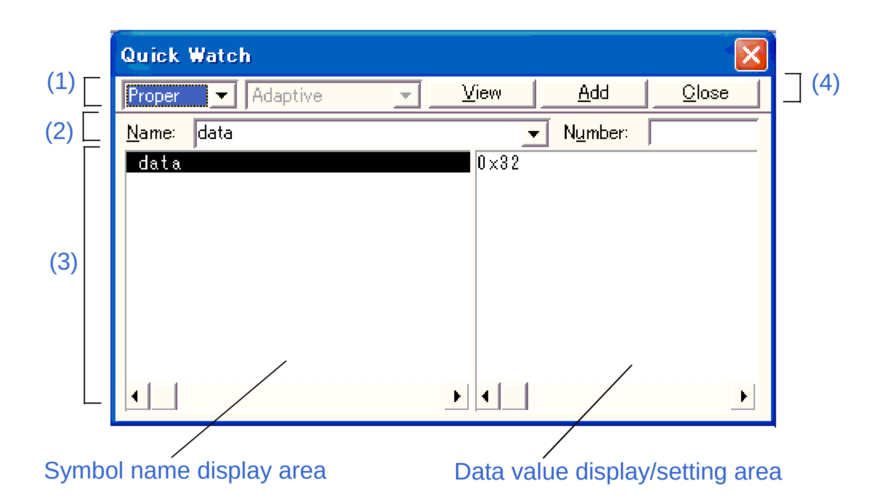
(在主菜单中选择同样的名称选项执行同样的操作)。

访问此变量时产生断点	使用选中的项创建一个可以读写的断点事件。
写入此变量时产生断点	使用选中的项创建一个可以写的断点事件。
从此变量读出时产生断点	使用选中的项创建一个可以读的断点事件。
清除	删除与选中项目相关的断点事件。
RRM 设置...	设置RRM功能的采样范围 打开RRM设置对话框**[IECUBE]**
事件信息	显示选中变量的事件信息，如果已设置事件，将打开事件对话框。
修改观察窗口...	修改选中的观察数据。 打开修改观察对话框。
删除观察窗口	删除窗口中选中的观察数据。
二进制	用二进制方式显示选择的行。
八进制	用八进制方式显示选择的行。
十进制	用十进制方式显示选择的行。
十六进制	用十六进制方式显示选择的行。
字符串	用字符串方式显示选择的行。
Proper合适的	以每个变量缺省值的方式显示选中的行。 根据调试选项对话框的设置显示符号(缺省)。
Byte字节	以8比特为单位显示选择的行
Half Word半字	以16比特为单位显示选择的行
Word字	以32比特为单位显示选择的行
Adaptive适应	用每个变量的缺省值显示选择的行(缺省) 对于C语言符号只有这一项是有效的选择。 根据调试选项对话框的设置显示汇编语言中符号。
Up向上	向上移动选择的行
Down向下	向下移动选择的行

快速观察对话框

这个对话框用来临时显示或修改指定的观察数据 (参考"5.6 观察窗口".)

图 6-28：快速观察对话框



- 打开
- 每个区域解释

打开

当源文件窗口或汇编窗口是当前窗口时，选择[View] 菜单中[Quick Watch...]选项，或者在窗口中单击<Quick...> 按钮。

每个区域解释

(1)　指定区域的显示

Display radix selection area显示数制选择区域	这个区域用来选择显示数制	
	Proper适合的	变量：显示每个变量缺省的值 符号：显示调试选项对话框中设置的数制
	Hex十六进制	用十六进方式显示选择数据
	Dec十进制	用十进制方式显示选择数据
	Oct八进制	用八进制方式显示数据
	Bin二进制	用二进制制方式显示数据
	String字符串	用字符串方式显示数据

Display size selection area显示大小选择区域	此区域用来选择显示区大小。 如果显示区大小固定，比如要显示C语言变量或者寄存器，无法修改。	
	Adaptive适应	变量：显示每个变量符号的缺省值 变量符号：以调试选项对话框中设置的长度显示数据
	Byte字节	以8比特为单位显示
	Half Word半字	以16比特为单位显示
	Word字	以32比特为单位显示

(2)　观察数据设置区域

Name:名字：	这个区域用来指定要显示的观察数据。 缺省条件下，显示调用此对话框的窗口中选中的字符串。如果必要，可以修改显示的字符串。最多可以记录16条输入的历史纪录。 如果此区域的内容被修改，点击<View>按钮，指定的数据可以在下部显示。
Number:数量：	这个区域用来指定要显示的数据个数(为空或者1到256) 如果此区域为空，数据显示为一个简单变量。如果指定了1个或者更多，数据在观察窗口中显示为一个数组变量。 如果显示了一个数组变量，数据前面将附加一个"+"前缀。双击"+"，所有的数据单元被扩展并且按照数据类型显示。(数据前面将附加一个"-"前缀。如果双击"-"，扩展显示被取消。) 如果要显示的数据个数是固定的，比如要显示的是C语言变量或者寄存器，指定的数据个数无效。

(3)　观察数据显示区域

左边的区域显示符号名称，右边的区域显示数据值。

Symbol name display area符号名称显示区域	这个区域用来显示观察数据(变量名字，符号名字和类型，以及结构或者联合的标记名字)。(参考"观察窗口"。)此区域不能被编辑。
Data value display/setting area数据值显示/设置区域	此区域用来显示和修改数据值。(参考"观察窗口"。)

(4)　功能按钮

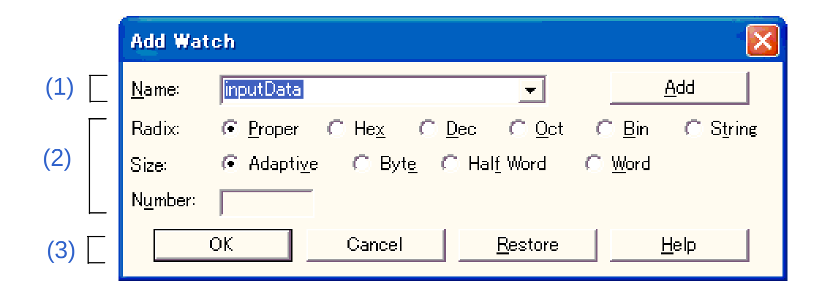
View查看	在下面的区域显示在(2)观察数据指定区域中指定的数据。
Add添加	添加在(2)观察数据指定区域中指定的数据到观察窗口。
Close关闭	关闭对话框。 没有写入目标存储器的数据被取消。

添加观察对话框

这个窗口用来登录观察数据，这些数据将在观察窗口中显示 (参考"5.6　观察功能".)

可以登录具有同一个名称的多个数据。

图 6-29：添加观察对话框



- 打开
- 每个区域解释

打开

选择[View]菜单中的[Add Watch...]选项，或单击观察窗口中<Add...> 按钮。

每个区域的解释

(1)　观察数据登录区域

Name:名称	这个区域用来将指定的符号添加到观察窗口中 (参考"表6-12　观察窗口输入格式".) 在缺省方式下，显示调用此对话框窗口中选中的字符串。 必要时，显示的字符串可以被修改。 如果没有选中的字符串，这个区域是空的。最多可以保留16个输入的历史记录
<Add>添加	添加数据到观察窗口。 这个对话框仍然保持打开状态。

表 6-12 观察窗口输入格式

-　C语言变量名	
变量表达式：变量名	
|变量表达式 [常量值 |变量名字]	数组单元
|变量表达式 .成员名字	结构/联合的实体成员
|变量表达式 ->成员名字	指针指示的结构/联合的成员
|*变量表达式	指针变量值
|&变量表达式	变量的地址
-..寄存器名称	
-..IOR名称, IOR 位名称	
-..立即数标号和地址	
-..寄存器 名字.bit	
-..IOR名字. bit	
-..标号名字.bit ,立即数.bit的地址	
-..范围指定	

指定范围下变量的处理方法如下所示。

表 6-13：指定范围下变量的处理方法

范围指定	程序名字	文件名字	函数名	变量名
prog$file#func#var	prog	file	func	var
prog$file#var	prog	file	global	var
prog$func#var	prog	global	func	var
prog$var	prog	global	global	var
file#func#var	current	file	func	var
file#var	current	file	global	var
func#var	current	current	func	var
var	current	current	current	var

(2)　显示方式改变区域

Radix:数制	这个区域用来选择显示的数制	
	Proper适合的	变量：显示每个变量的缺省值。 符号：使用在调试选项对话框中设置的数制显示数据。
	Hex16进制	显示十六进制的数。
	Dec10进制	显示十进制的数。
	Oct8进制	显示八进制的数。
	Bin2进制	显示二进制的数。
	String字符串	显示字符串。
Size: 大小	此区域用来选择显示区域大小。如果显示区域大小固定，比如要显示的是C语言变量或者寄存器，无法修改。	
	Adaptive适应	变量：显示每个变量缺省的值。 符号：显示调试选项对话框中设置长度的数据。
	Byte字节	以8比特为单位显示。
	Half Word半字	以16比特为单位显示。
	Word字	以32比特为单位显示。
Number:数目	这个区域用来指定要显示的数据个数(为空或者1到256) 如果此区域为空，数据显示为一个简单变量。如果指定了1个或者更多，数据在观察窗口中显示为一个数组变量。 如果显示了一个数组变量，数据前面将附加一个"+"前缀。双击"+"，所有的数据单元被扩展并且按照数据类型显示。(数据前面将附加一个"-"前缀。如果双击"-"，扩展显示被取消。) 如果要显示的数据个数是固定的，比如要显示的是C语言变量或者寄存器，指定的数据个数无效。	

(3)　功能按钮

OK确认	增加指定的数据到观察窗口。 关闭对话框。
Cancel取消	取消修改并关闭对话框。
Restore恢复	恢复对话框打开前的状态。
Help帮助	显示对话框在线帮助文件。

改变观察对话框

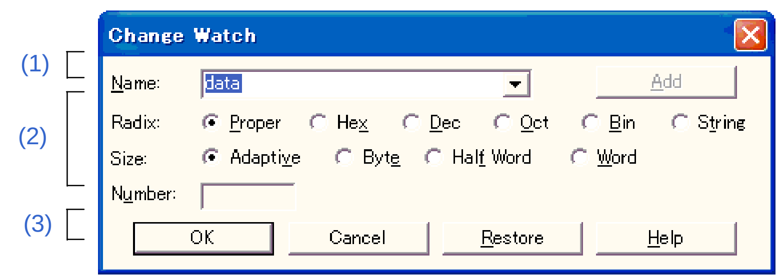
这个窗口用来改变观察窗口中选定行中的数据 (参考"5.6 观察功能".)

当链式打开一行时，如数组成员，结构和联合的元素是不能被改变的。

当观察数据改变时，选定行的内容被新数据代替

即使会导致和已有文件重名，符号名称仍然可以改变。

图 6-30：改变观察对话框



- 打开
- 每个区域解释

打开

当观察窗口是当前窗口时，选择[View]菜单中[Change Watch...]选项.

每个区域解释

(1)　观察数据改变区域

Name:名称	这个区域用来修改观察窗口中选中的行的符号名称(参考表6-12"观察窗口输入格式")。 即使可能导致和使用中的数据重名，符号名字也可以被修改。 缺省条件下，显示调用此对话框的窗口中选中的字符串。最多可以记录16条输入的历史纪录。
<Add>添加	无法选择。

(2) 显示方式改变区域

Radix:数制	这个区域用来改变在观察窗口中选择行的显示数制。	
	Proper适合的	变量：显示每个变量的缺省值 符号：使用调试选项对话框中所设置的数制显示数据。
	Hex16进制	显示十六进制数
	Dec10进制	显示十进制数
	Oct8进制	显示八进制数
	Bin2进制	显示二进制数
Size:大小	这个区域用来修改观察窗口中被选择的一行显示的长度。 如果显示长度是固定的，比如显示的是C语言变量或者寄存器被，显示长度不能够被改变。	
	Adaptive	变量：显示每个变量的缺省值。 符号：以调试选项对话框中设置的长度显示数据。
	Byte	以8比特为单位显示
	Half Word	以16比特为单位显示
	Word	以32比特为单位显示
Number:数目	这一个区域用来修改观察窗口被选择的一行显示的数据个数。(空白或者1 to 256之间)。 如果这一个区域是空白，数据被显示为一个简单的变量。 如果指定为1或者更多，数据被显示为观察窗口的一个数组变量。 如果显示一个数组变量，"+" 被前缀到数据。 双击 "+"，将扩展显示所有类型数据("-"被前缀到扩展的数据。 如果双击"-"，扩展显示被取消)。 如果显示的数据个数是固定的，诸如显示一个 C语言变量或者寄存器，指定数据的个数是无效的。	

(3) 功能按钮

OK确认	用指定的数据代替观察窗口中选中行上的数据，然后关闭对话框。
Cancel取消	关闭对话框。
Restore恢复	恢复到打开对话框前以前的状态。
Help帮助	显示对话框在线帮助文件。

局部变量窗口

这个窗口用来实现对当前函数中局部变量进行修改和显示。(参考"5.6.2 显示和改变局部变量".)

这个窗口和堆栈窗口用跳转函数关联，当跳转到源文件窗口时，将显示跳转函数的局部变量

在此窗口中使用上下文菜单和功能按钮还可以执行其它的操作功能

图 6-31：局部变量窗口

Local Variable (sample)

Refresh　Close

num_idx　0x00000001

(1)

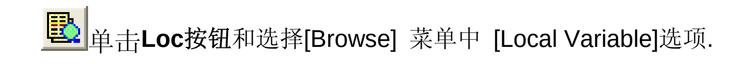
(2)

局部变量名显示区域

局部变量植显示/设置区域

- 打开
- 每个区域解释
- [View]菜单 (局部变量窗口专用功能)
- 上下文菜单

打开



单击**Loc按钮**和选择[Browse] 菜单中 [Local Variable]选项.

每个区域解释

(1)　局部变量显示/修改区域

局部变量名称显示区域	这个区域用来显示局部变量的名称 (参考观察窗口中的"符号名称显示区域") 可以显示自动、内部静态和寄存器变量。 在这个窗口中自动显示当前窗口中的局部变量。这个区域不能编辑。
局部变量值显示/设置区域	这个区域用来显示和修改局部变量的数值。 (参考观察窗口中"数据值显示/设置区域")

(2) 功能按钮

Refresh刷新	以最新观察数据刷新窗口内容。
Close关闭	关闭窗口

[View] 菜单 (局部变量专用功能)

当这个窗口是当前窗口时，下列的条目被增加到[View] 菜单中。

Bin2进制	显示二进制数
Oct8进制	显示八进制数
Dec10进制	显示十进制数
Hex16进制	显示十六进制数
String字符串	显示字符串
Proper合适的	显示每个变量的缺省值(缺省)

上下文菜单

菜单项对于选择的行或者项有效，而不是鼠标所点击的位置。

(在主菜单中选择同样的名称选项执行同样的操作)。

Add Watch...添加观察窗	打开添加观察对话框。
Bin2进制	以二进制方式显示选择的行。
Oct8进制	以八进制方式显示选择的行。
Dec10进制	以十进制方式显示选择的行。
Hex16进制	以十六进制方式显示选择的行。
String字符串	以字符串方式显示选择的行。
Proper合适的	用每个变量的缺省值显示选中的行。 更具调试选项对话框设置的方式显示符号(缺省)。

堆栈窗口

这个窗口用来显示或修改当前用户程序的堆栈内容(参考"5.6.7 堆栈跟踪显示功能".)

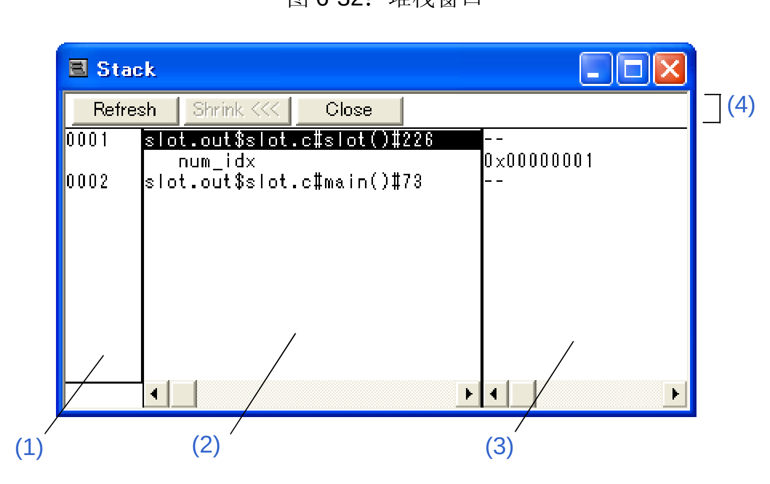
利用跳转功能可以实现跳转至与堆栈内容对应的窗口。

在该窗口中还可以利用上下文菜单、功能按钮实现其他功能。

注意事项:　如果一个函数没有创建堆栈结构，堆栈跟踪显示功能可能不能正确操作。

备注:　[ERROR] 可能在程序的预处理或后处理中出现错误。

图 6-32：堆栈窗口



- 打开
- 每个区域解释
- [View]菜单 (堆栈窗口专用功能)
- 上下文菜单

打开

单击**Stk** 按钮或选择 [Browse] 菜单中 [Stack Trace]选项.

每个区域解释

(1)　堆栈结构数目显示区域

这个区域用来设置堆栈结构的序号和显示堆栈的内容。

堆栈结构的序号从自然数1开始。堆栈嵌套越浅数字越高。这意味着如果一个函数的堆栈序号比另一个函数大1，则前者调用后者。

(2)　堆栈结构内容显示区域

这个区域显示堆栈结构的内容。

它显示函数或者局部变量的名称。注意：这个区域不能进行编辑

如果堆栈内容包含一个函数	将显示如下： **[程序名称$文件名称#函数名称(参数列表) #行号]** 如果双击这一行，所执行的操作和跳转功能中跳转到源文件窗口相同(例如，局部变量窗口中将显示程序执行跳转到的函数中的局部变量)。如果函数包含一个局部变量，在随后的下一行中将显示此局部变量。
如果堆栈内容包含一个局部变量	显示它的类型和名称(参考"观察窗口"。)　注意不显示内部静态变量和寄存器变量

(3)　堆栈内容显示/设置区域

这个区域用来显示和改变堆栈内容。

如果堆栈内容是一个函数	显示"--"函数不能被修改。
如果堆栈内容是一个局部变量	显示这个变量值(参考"观察窗口".)

(4)　功能按钮

Refresh刷新	以最新数据更新窗口内容。
Shrink <<<收缩	收缩选择函数中的局部变量列表。
Expand >>>扩展 (when the <Shrink<<<> button is clicked)当单击了<Shrink<<<>按钮	显示选择函数中的局部变量列表。
Close关闭	关闭此窗口

[View] 菜单 (堆栈窗口专用功能)

当这个窗口是当前窗口，下列条目将添加到 [View] 菜单.

Bin	显示二进制数
Oct	显示八进制数
Dec	显示十进制数
Hex	显示十六进制数
String	显示字符串
Proper	显示每个变量的缺省值(缺省)

上下文菜单

Bin	显示二进制数
Oct	显示八进制数
Dec	显示十进制数
Hex	显示十六进制数
String	显示字符串
Proper	显示每个变量的缺省值(缺省)
Source Text	显示从光标处的数据值所指定的跳转地址相应的源文本行和源文件行。(参考"5.16.2 跳转功能".) 如果跳转的目的地址没有行信息，将不能执行跳转。 打开源文件窗口。 如果打开了一个激活的源文件窗口，这个窗口在最前面显示(因此可以对它进行操作).
Assemble	从光标位置处的数据值指定的跳转目的地址处开始反汇编并且显示。(参考"5.16.2跳转功能") 打开汇编窗口。 如果打开了一个激活的汇编窗口，这个窗口在最前面显示(因此可以对它进行操作)..
Memory	从光标位置处的数据值指定的跳转目的地址处开始显示存储内容。(参考"5.16.2跳转功能") 打开存储器窗口。 如果打开了一个激活的存储器窗口，这个窗口在最前面显示(因此可以对它进行操作)..

存储器窗口

这个窗口用来显示和改变存储器内容（参考"5.7 存储器管理功能"。）在这个窗口中利用上下文菜单、功能按钮还可以执行其他操作。而且，这个窗口具有静止和激活两个状态。当窗口处于激活状态下，具有关联窗口的跟踪结果[IECUBE]和跳转功能 。（参考"5.16 每个窗口的公共功能"。）

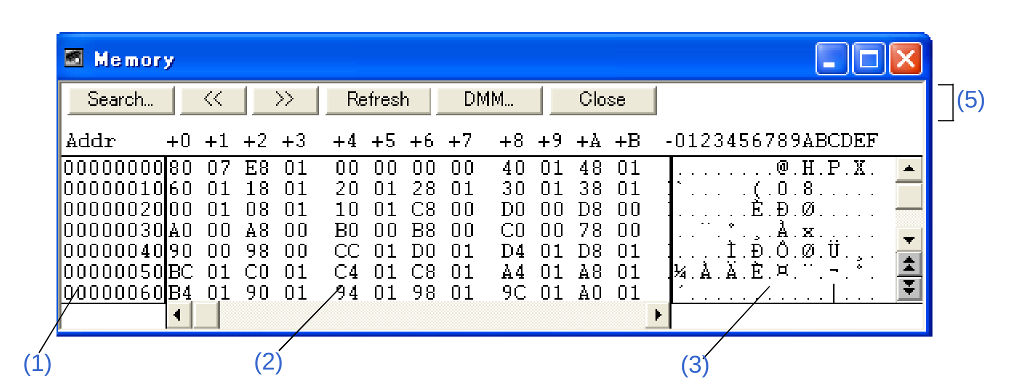
备注1: 采用不同的颜色显示存储器访问状态（读，写，读&写）(参考"5.7.2 访问监视器功能 [IECUBE]".) [IECUBE]

备注2: 窗口打开时显示的起始地址如下：

第一次：从RAM区的起始地址开始显示。

第二次及以后：从关闭激活状态窗口的地址开始显示（如果一个激活窗口从未关闭，从第一次显示的起始位置开始显示).

图 6-33：存储器窗口



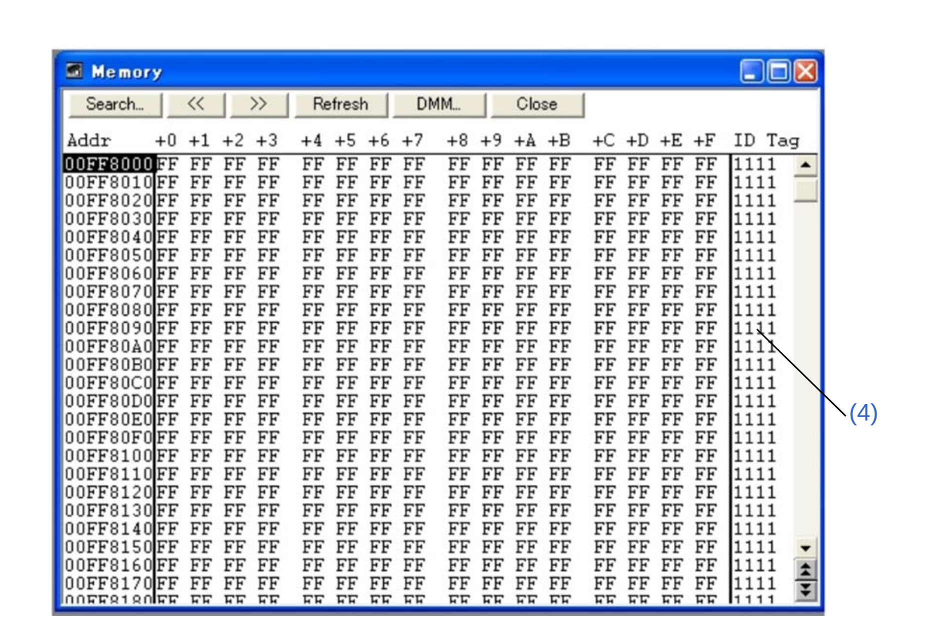
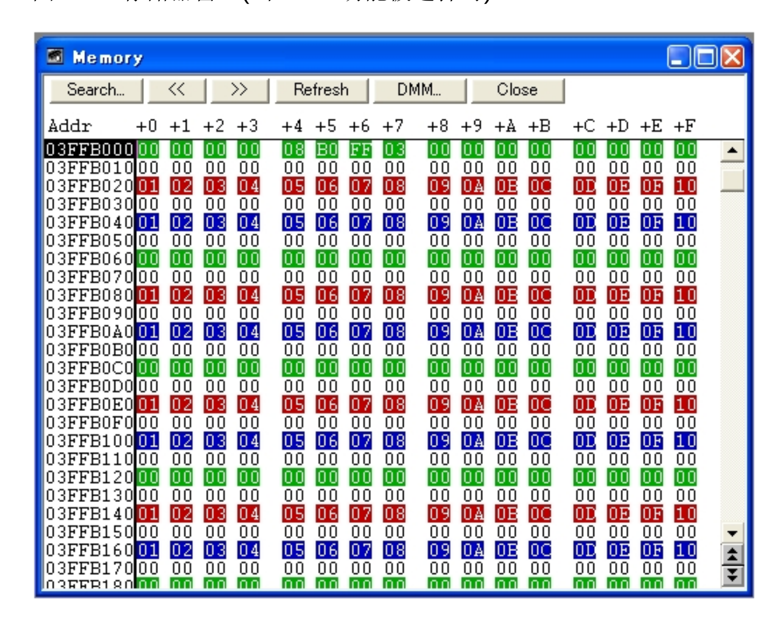


图 6-34 存储器窗口(当 RRM 功能被选择时)



- 打开
- 每个区域解释
- [View] 菜单(存储器窗口专用菜单)
- 上下文菜单

打开

单击"**Mem**"按钮，或选择[Browse]菜单中[Memory]选项。

每个区域解释

(1) Addr

这个区域显示存储器地址。

通过单击此区域相应的行可以选择任意一个地址。选择的地址将高亮显示。

(2)　+0 +1 +2....

这个区域用来显示和改变存储器的内容，同时显示访问状态 (参考"5.7.2访问监视器功能[IECUBE]".)

显示		当显示信息保存到观察文件中时的替代符号	含义
00	Green	R	读
00	Red	W	写
00	Blue	A	读&写
00	无高亮颜色	None	超出测量范围

可以直接输入修改存储器的内容。修改的位置标记为红色，当按下回车键时修改的内容将写入目标存储器。先前的内容可以通过ESC键取消。一次最多可指定256 字节。

备注:　通过单击<DMM...>按钮打开DMM对话框可以在用户程序执行期间修改存储器的内容。

(3) 0 1 2 3....

这个区域通过ASCII码方式修改和显示存储器的内容。

通过选择[View] 菜单的[Ascii] 选项可以显示这个区域

这个区域的数据可以采用和存储器显示区域相同的方法进行修改。

修改方法和 (2) +0 +1 +2.....相同。

备注:　当改变显示地址时，ASCII显示区域的光标位置是不同步的。

(4)　ID标记

这个区域用来显示闪存中的ID标记。

ID标记是一个比特，用来指示闪存中的一个字，用来检测电源故障。当它为"0"，表示这个字中的数据写入正常。

这个区域的设置可以在"0" 和"1"之间改变。

可以通过选择[View]菜单(缺省：隐藏)中的[ID tag]选项来切换是否显示这个区域。这个区域不能和"(3) 0 1 2 3....". 区域同时显示。

(5)　功能按钮

Search...	打开存储器搜索对话框，从显示或未显示的存储器内容中查找字符串。 在存储搜索对话框中选择一个显示的数据(一个存储器值)，作为搜索值。如果存储搜索对话框打开时没有指定数据，从键盘指定数据。 搜索到的数据在存储区窗口中高亮显示。
<<	搜索符合在存储区搜索对话框中设置的搜索条件的存储内容，从光标位置处向前(屏幕中向上)。 在搜索中这个按钮将显示为<Stop>。
>>	搜索符合在存储区搜索对话框中设置的搜索条件的存储内容，从光标位置处向前(屏幕中向上)。 在搜索中这个按钮将显示为<Stop>。
Stop(searching)	停止搜索。
Refresh	以最新数据更新窗口内容。
DMM...	打开DMM对话框。
Close	关闭此窗口。

[View] 菜单(存储器窗口专用功能)

当存储器窗口处于激活状态时，下列内容将加入到 [View] 菜单中。

Bin	显示二进制数
Oct	显示八进制数
Dec	显示十进制数
Hex	显示十六进制数(缺省).
Nibble	以4比特为单位显示
Byte	以8比特为单位显示(缺省).
Half Word	以16比特为单位显示
Word	以32比特为单位显示
Ascii	选择是否显示ASCII字符。 选中：显示 清除：隐藏(缺省)
ID tag	切换是否显示数据闪存ID标记。 选中：显示 未选中：隐藏 (缺省) ID显示和ASCII显示不能同时进行。
Little Endian	以Little Endian显示(缺省)
Big Endian	以Big Endian显示
Access Monitoring **[IECUBE]**	设置访问监视功能

Clear **[IECUBE]**	通过访问监视器功能清除显示颜色。
Accumulative **[IECUBE]**	使能/禁止访问状态的累计显示(存储器内容修改). 选中：累计显示存储器内容改变 未选中：只显示以前修改的存储器改变值

上下文菜单

菜单项对于选择的行或者项有效，而不是鼠标所点击的位置。

(在主菜单中选择同样的名称选项执行同样的操作)。

Move...	移动显示位置 打开地址移动对话框
RRM Setting...	打开RRM设置对话框　**[IECUBE]**
Bin	显示二进制数
Oct	显示八进制数
Dec	显示十进制数
Hex	显示十六进制数(缺省).
Nibble	以4比特为单位显示
Byte	以8比特为单位显示(缺省).
Half Word	以16比特为单位显示
Word	以32比特为单位显示
Ascii	选择是否显示ASCII字符 选择：显示 清除：隐藏(缺省)
ID tag	切换是否显示数据闪存ID标记 选中：显示 未选中：隐藏 (缺省) ID标记显示和ASCII显示不能同时执行
Clear Access Monitoring **[IECUBE]**	通过访问监视器功能清除显示颜色。
Accumulative **[IECUBE]**	使能/禁止访问状态的累计显示(存储器内容修改)。 选中：累计显示存储器内容改变 未选中：只显示从上次更新以后所做的修改。

存储器搜索对话框

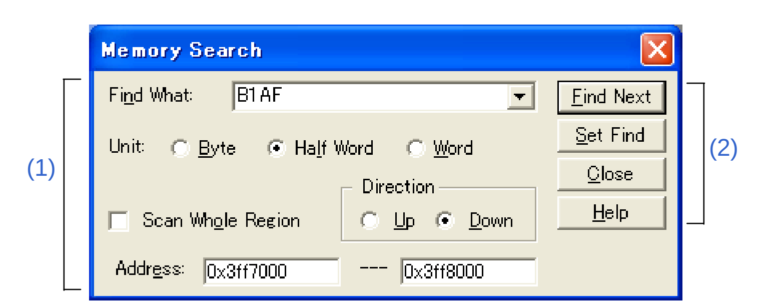
这个对话框用来在光标所在的存储器窗口中实现存储器内容搜索 (参考"5.7 存储器操作功能".)

如果光标停在存储器窗口中的存储器显示区域，指定的数据按二进制数据串处理，如果光标位于ASCII显示区域，指定的数据按照ACSII数据串处理，同时对相应区域的内容进行搜索。

设置搜索项目并单击<Find Next>按钮，开始执行搜索操作。通过在存储器窗口中单击<Set Find>按钮，在搜索中可以使用方向按钮 ("<<" 和">>")。

注意事项: 非映射，周边I/O寄存器和I/O保护区域不会被搜索.

图 6-35：存储器搜索对话框



- 打开
- 每个区域解释

打开

当存储器窗口是当前窗口时，选择[View]菜单中[Search...]选项，或在同一窗口中单击<Search...>按钮。

每个区域解释

(1) 搜索条件指定区域

区域		说明
Find What:搜索内容	此区域用来指定要搜索的数据。(最多256个字符。)在缺省条件下，显示在调用此对话框的窗口中选中的字符串。如果必要，可以修改显示的字符串。最多可以记录16个输入的历史纪录。	
	当在存储显示区域搜索时	.最多可指定16个数据项。使用一个"空字符"分隔数据。
	当在ascii区域搜索时	最多可以指定256个字符。数据中的"空字符"被当作空字符对待。
Unit:单位	这个区域用来指定在存储显示区中要搜索的数据的比特数。	
	Byte	按照8比特数据搜索数据(缺省)。
	Half Word	按照16比特数据搜索数据。
	Word	按照32比特数据搜索数据。
Scan Whole Region搜索整个区域	搜索整个指定区域时需选中此项。	
Direction方向	这个区域用来指定搜索的方向	
	Up	前向搜索。从光标当前位置开始向前(在屏幕中向上)搜索数据。
	Down	后向搜索。从光标当前位置开始向前(在屏幕中向下) 搜索数据。
Address:地址	此区域用来指定要搜索的地址范围。 输入数值缺省为十六进制。也可以使用一个符号或者表达式指定地址。(参考 "表5-6 指定符号"。)	

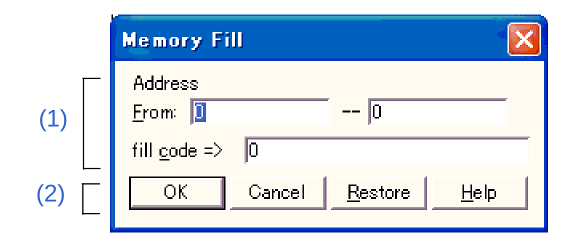
(2) 功能按钮

按钮	说明
Find Next查找下一个	根据给定条件搜索指定数据 如果搜索到指定的字符串，将高亮显示。再次点击此按钮，继续搜索。
Set Find搜索设置	设置指定的条件为搜索条件然后关闭对话框
Stop (searching)停止(搜索中)	停止搜索.
Close关闭	关闭此对话框(搜索期间，这个按钮将显示为<Stop>)
Help帮助	显示对话框在线帮助文件

存储器填充对话框

这个对话框用来对存储器窗口中内容进行填充 (填充代码). (参考"5.7 存储器填充对话框".)

图 6-36 存储器填充对话框



- 打开
- 每个区域解释

打开

选择[Edit]菜单[Memory]中的[Fill...]选项。

每个区域解释

(1) 存储器填充范围指定

Address	这个区域用来指定填充范围和填充代码	
	From:	指定填充范围 (起始地址 – 终止地址). 输入数据缺省为16进制。也可以用符号或者表达式指定地址(参考"表5-6 指定符号".)
	fill code =>	指定填充在"From:"中指定的范围所使用的数据 (填充代码)。 可以指定多至16个二进制数据串 (字节数据串)。 每个数据通过"空字符"分隔。

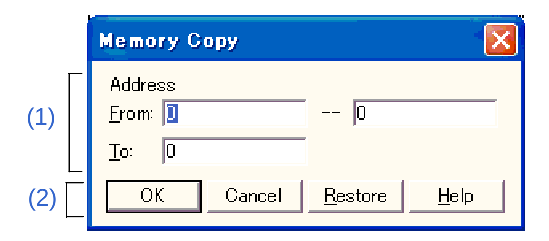
(2) 功能按钮

OK	根据给定的条件填充相应的数据。
Stop (filling)	停止填充。
Cancel	关闭对话框。(在填充期间，这个按钮将显示为<Stop>。)
Restore	恢复到对话框打开之前的状态。
Help	显示对话框在线帮助文件。

存储器复制对话框

这个窗口用来复制存储器窗口中的内容（参考"5.7 存储器操作功能".)

图 6-37：存储器复制对话框



- 打开
- 每个区域解释

打开

选择 [Edit] 菜单中[Memory]下的 [Copy...]选项。

每个区域解释

(1) 复制指定的区域

Address地址	这一个区域用来指定数据复制的源和目标地址。 输入数据缺省为16进制。可以用符号或者表达式指定地址(参考"表5-6 指定符号".)	
	From:	指定复制源地址范围(起始地址—终止地址)。
	To:	指定复制目的的起始地址。

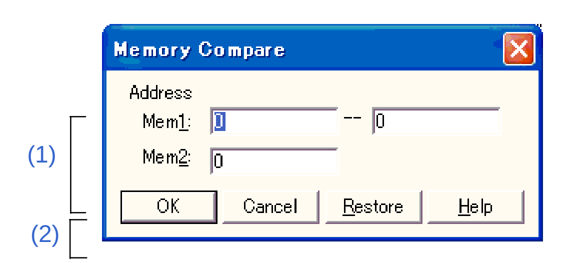
(2) 功能按钮

OK	根据给定的条件复制存储器内容。
Stop (copying) 停止(复制中)	停止复制。
Cancel取消	关闭对话框（在复制期间，这个按钮将显示为<Stop>。）
Restore恢复	恢复对话框打开前的状态。
Help帮助	显示对话框在线帮助文件。

存储器比较对话框

这个对话框用来实现对存储器窗口中内容的比较。(参考"5.7 存储器操作窗口".)

图 6-38：存储器比较对话框



- 打开
- 每个区域的比较

打开

选择[Edit] 菜单中[Memory] 的 [Compare...] 选项。

每个区域解释

(1)　比较范围指定区域

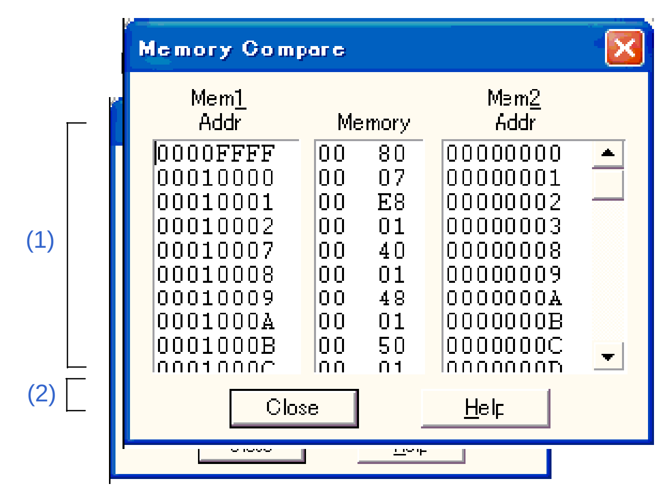
Address地址	这一个区域被用来指定数据比较的源和目标地址。 输入数据缺省为16进制。 可以用符号或者表达式指定地址 (参考"表5-6 指定符号"。)	
	Mem1:	指定比较的源地址范围 (起始地址—终止地址)。
	Mem2:	指定目的比较地址的起始地址。

(2)　功能按钮

OK	根据给定的条件进行存储器内容比较。 如果比较内容没有差别，将显示" Wf200: 没有发现差别." 如果比较的内容有差别，将打开存储器比较结果对话框。
Stop (comparison) 停止(比较)	停止存储器比较
Cancel取消	关闭对话框 (在比较期间，这个按钮被将显示为<Stop>)
Restore恢复	恢复打开对话框之前的状态
Help帮助	显示对话框在线帮助文件

存储器比较结果对话框

当对存储器比较对话框中的存储器内容进行比较时，如果内容有任何不同，将显示这个对话框。(参考"5.7 存储器操作功能"。)



- 每个区域的解释

每个区域的解释

(1) 比较结果显示区域

这个区域显示存储器比较的结果。只显示比较结果不同之处。

Mem1 Addr存储区1地址	显示发现不同内容处的源地址。
Memory存储区	显示发现不同内容的数据。 (左：比较源数据，右：比较目的数据)。
Mem2 Addr存储区2地址	显示比较时发现不同处的目的地址。

(2) 功能按钮

Close关闭	关闭对话框
Help帮助	显示对话框在线帮助文件

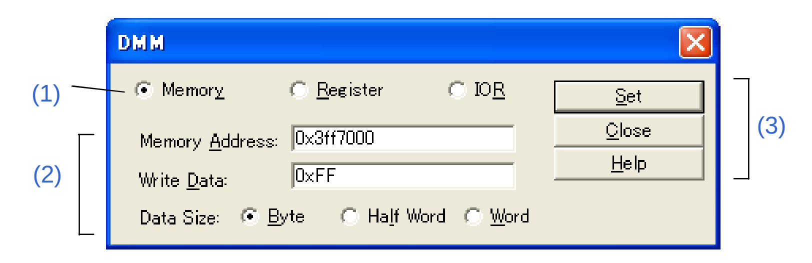
DMM 对话框

这个对话框用来设置DMM(动态存储器修改)的地址和数据。(参考"5.14DMM功能")

通过DMM功能可以实现在用户程序执行期间重新写入存储器的内容。

注意事项1:　当连接IECUBE时，执行伪DMM功能。对于伪DMM功能，当用户程序执行写操作瞬间将产生一个断点。

图 6-40：DMM 对话框（例：当"Memory" 被选中时）



- 打开
- 每个区域的解释

打开

选择[Edit] 菜单中 [DMM...]选项，或在存储器窗口/ 寄存器窗口/ IOR窗口单击<DMM...> 按钮.

每个区域的解释

(1) DMM目标选择区域

这个区域用来为DMM选择目标系统。通过选择选项按钮可以修改(2)DMM设置选择区域(2)中显示的内容。

Memory存储器	对存储器执行DMM功能
Register寄存器	对寄存器执行DMM功能
IOR	对IOR执行DMM功能

备注:　如果通过存储器窗口、寄存器窗口或IOR窗口打开DMM对话框，将同时选中相应的选项按钮。

(2)　DMM设置区域

(a) 当选择存储器时

Memory Address: 存储器地址	这个区域用来指定写入数据的存储器的地址。 输入数据缺省为十六进制。也可以用一个符号或者表达式指定地址 (参考 "表 5-6 指定符号".)	
Write Data:写入数据	这个区域用来指定要写入在"存储器地址"中指定的内存地址处的数据。	
Data Size:数据长度	这个区域用来指定在"写入数据："中写入数据的长度。	
	Byte:	按8比特方式写入数据
	Half Word:	按16比特方式写入数据
	Word:	按32比特方式写入数据

(b)　当选择寄存器时

Register Name:寄存器名	这个区域用来指定需要写入数据的寄存器的名称。 区别大小写。即可以指定功能名称又可以指定绝对名称。
Write Data:写入数据	这个区域用来指定要写入寄存器的数据，此寄存器在"寄存器名称"中指定。

(c)　当选择IOR时

IOR Name: IOR名字	这个区域用来指定要写入数据的IOR名称。 区分大小写。不能指定只读的特殊功能寄存器(SFRs)。
Write Data:写入数据	这个区域用来指定要写入寄存器的数据，此寄存器在"IOR名称"中指定。

(3)　功能按钮

Set设置	根据给定的条件写入数据
Close关闭	关闭对话框。
Help帮助	显示对话框在线帮助文件

寄存器窗口

这个窗口用来改变和显示寄存器内容 (程序寄存器/系统寄存器). (参考"5.8 寄存器操作功能".)

在这个窗口中可用上下文菜单、功能按钮，等等实现其它操作。

这个窗口的每个区域都是跳转功能的跳转指针。

图 6-41：寄存器窗口

Register

Refresh　DMM...　Close　(2)

(1)

r0(zero)	00000000
r1	FFFFFFF8
r2	00000000
r3(sp)	03FFB470
r4(gp)	0400310C
r5(tp)	00000000
r6	00000002
r7	00000000
r8	00000000
r9	00000000
r10	00000002
r11	00000000
r12	00000000
pc	00000694
eipc	00000000
+eipsw	00000000
fepc	00000000
+fepsw	00000000
+ecr	00000000
+psw	00000000
ctpc	0000080A
+ctpsw	00000000
dbpc	00000000
+dbpsw	00000000
ctbp	000003C8

- 打开
- 每个区域解释
- [View] 菜单(寄存器窗口专用功能)
- 上下文菜单

打开

单击Reg 按钮或选择[Browse]菜单中 [Register]选项。

每个区域解释

(1)　寄存器数值显示/修改区域

左边区域显示寄存器名称，右边区域显示寄存器数值。

寄存器数值可以直接输入进行修改。修改的区域显示为红色，当按回车键时，修改的内容将写入目标存储器中。

先前的数值可通过ESC键取消。上面区域显示程序寄存器，下面区域显示系统寄存器。

程序寄存器显示/修改区域	这个区域用来显示和修改程序寄存器
系统寄存器显示/修改区域	这个区域用来显示和修改系统寄存器。 双击"+", 显示标志名和标志值(第一个字符从"+"变为"-")。 通过双击"-"取消扩展显示 (第一个字符从"-" 变为"+").

注意事项:　　当用户键入非法数值时，寄存器发生溢出。寄存器的值将更新为0xFFFFFFFF。

(2)　功能按钮

Refresh刷新	以最新数据更新窗口内容
DMM...	打开DMM对话框
Close关闭	关闭此窗口

[View] 菜单 (寄存器窗口专用功能)

当寄存器窗口是激活状态时，下列的内容将添加到 [View] 菜单中。

Bin	显示二进制数。
Oct	显示八进制数。
Dec	显示十进制数。
Hex	显示十六进制数(缺省)。

Pick Up提取	只显示寄存器选择对话框选择的寄存器。
Select...选择	打开寄存器选择对话框。

上下文菜单

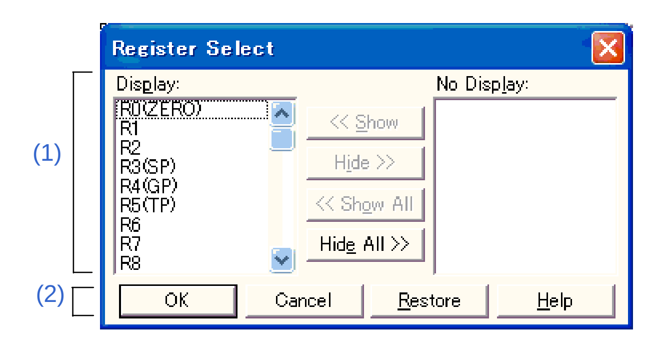
Add Watch...添加观察窗口	登录一个选择的字符串到观察窗口。打开添加观察对话框。
Bin	显示二进制数。
Oct	显示八进制数。
Dec	显示十进制数。
Hex	显示十六进制数(缺省)。

Pick Up提取	只显示寄存器选择对话框选择的寄存器。
Select...选择	打开寄存器选择对话框。

寄存器选择对话框

这个窗口用来显示那些不在寄存器窗口中显示的寄存器 (参考"5.8 寄存器操作功能".)

图 6-42：寄存器选择对话框



- 打开
- 每个区域解释

打开

当寄存器窗口是当前窗口时，选择[View] 菜单中[Select...]选项.

每个区域的解释

(1) 显示寄存器选择区域

Display:显示	在寄存器窗口显示寄存器。	
Button按钮	下列按钮用来改变要显示的寄存器。 按住Ctrl或Shift键，通过单击上面的按钮可以选择两个或多个寄存器。	
	<< Show	把"不显示"列表中选择的寄存器移动到"显示"。
	Hide >>	把"显示"列表中选择的寄存器移动到"不显示"。
	<< Show All	移动所有寄存器到"显示"。
	Hide All >>	移动所有寄存器到"不显示"。
No Display:不显示	在寄存器窗口中不显示寄存器。	

(2) 功能按钮

OK确认	在寄存器窗口中反映此对话框中选择的内容并关闭对话框。
Cancel取消	取消修改关闭对话框。
Restore恢复	恢复到在对话框打开之前的状态。
Help帮助	显示对话框在线帮助文件。

IOR 窗口

这个窗口用来显示和修改周边I/O寄存器的内容和那些在增加I/O端口对话框中登录过的I/O端口。(参考"5.8 寄存器操作功能"。)

在这个窗口中通过上下文菜单、功能按钮等等可以实现其它多项操作。

注意事项1: 但是， 只读周边 I/O 寄存器和I/O端口的值不能被修改。而且，那些可能导致设备操作的外围I/O寄存器和外围I/O 端口是读保护的，因此这些端口不能被读出。可以从上下文菜单中选择这些寄存器的读、选择和选择执行[强制读]。

注意事项2: 在用户程序执行期间，IOR的内容在每个扩展对话框中设置的采样时间更新。(参考"5.13RRM功能"). 但是，当RAM监视功能关闭时，将不能更新。 **[IECUBE]**。

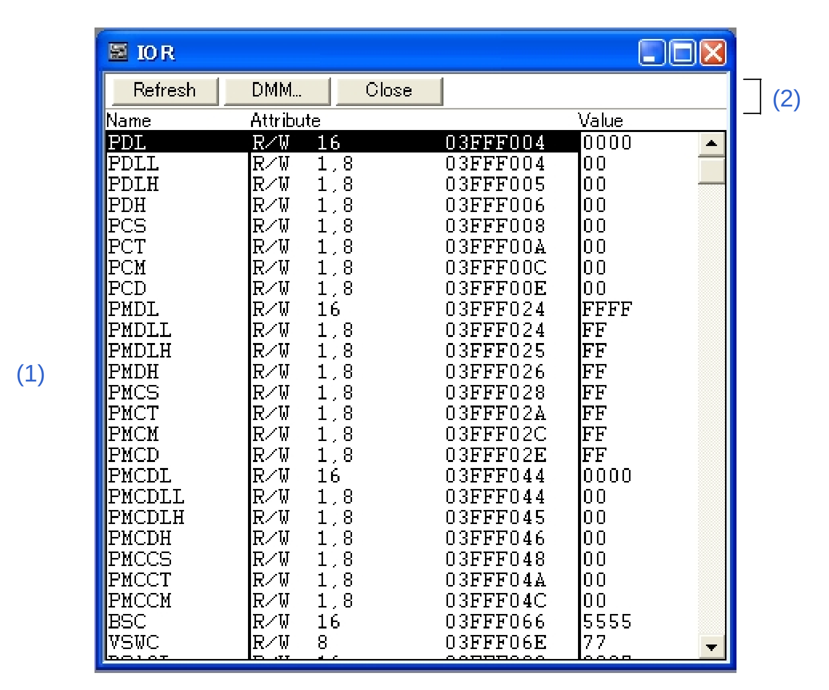
备注1: 当窗口时打开时，显示的起始位置如下：

第一次：从I/O 寄存器的最小地址开始显示。

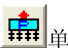
第二次和以后：从上次窗口关闭时第一个周边I/O寄存器开始显示。

备注2: 如果器件支持可编程I/O寄存器并在配置对话框中设置了可编程I/O寄存器区域，可编程I/O寄存器和扩展周边I/O区域也可以显示。如果一个I/O端口地址值已经定义，这个I/O端口名被显示为浅色。

图 6-43：IOR 窗口



打开

单击**IOR按钮**或选择[Browse]菜单中 [IOR]选项.

每个区域解释

(1) IOR显示/修改区域

Name名称	这个区域用来显示周边I/O寄存器和I/O端口的名称。 如果没有定义I/O端口地址值，I/O端口名称以浅色显示。	
Attribute属性	这个区域用来显示周边I/O寄存器和I/O端口的属性. 这个区域用来显示读/写属性，访问类型，同时从左边显示绝对地址。 当显示周边位I/O寄存器时，将同时显示位偏移量。 通过选择[View]菜单中的[Attribute].选项，指定这个区域是否显示。	
	读/写特性	
	R	只读
	W	只写
	R/W	读/写
	*	通过仿真寄存器读取此寄存器以防止读寄存器时器件操作。 执行[View]菜单中的[Compulsion Read]功能，直接读取周边I/O寄存器这个属性。 即使对于只写的周边I/O寄存器，也可通过仿真寄存器实现读操作 。但是某些器件不支持这种功能。
	访问类型	
	1	可以按比特访问
	8	可以按字节访问
	16	可以按半字访问
	32	可以按字访问

<table>
<tr><td rowspan="4">Value值</td><td colspan="2">这个区域用来显示和修改周边I/O寄存器和I/O端口的内容。
通过直接的输入改变数值。被改变的位置显示为红色，当按回车键时，修改的内容写入目标存储器。
通过ESC键可以取消先前的输入数值。
注意具有只读特性的周边I/O寄存器和I/O端口内容不能修改。
通过上下菜单中的[Compulsion Read]可以读取读保护的周边I/O寄存器值和I/O端口的内容.
根据下面选择的属性不同，显示的内容也将不同，如下所示：</td></tr>
<tr><td>全黑</td><td>只读或者读/写</td></tr>
<tr><td>--</td><td>只写</td></tr>
<tr><td>**</td><td>如果读将改变数值</td></tr>
</table>

(2)　功能按钮

Refresh刷新	以最新数据更新窗口内容。
DMM...	打开DMM对话框
Close关闭	关闭此窗口.

[View] 菜单(IOR 窗口专用功能)

当这个窗口是当前窗口时，下面的项目可以增加到[View]菜单中。

Bin	显示二进制数
Oct	显示八进制数
Dec	显示十进制数
Hex	显示十六进制数(缺省)
Sort By Name	按字母表顺序显示
Sort By Address	按地址顺序显示 （缺省）
Unsort	不排序
Attribute	切换打开/关闭显示属性。
Pick Up	只显示IOR选择对话框中选择的寄存器。
Select...	打开IOR选择对话框。
Compulsion Read	强制读取那些为避免数值改变而禁止读取的周边I/O寄存器，或者在添加I/O端口对话框中添加的I/O端口和I/O保护区域的数据。

上下文菜单

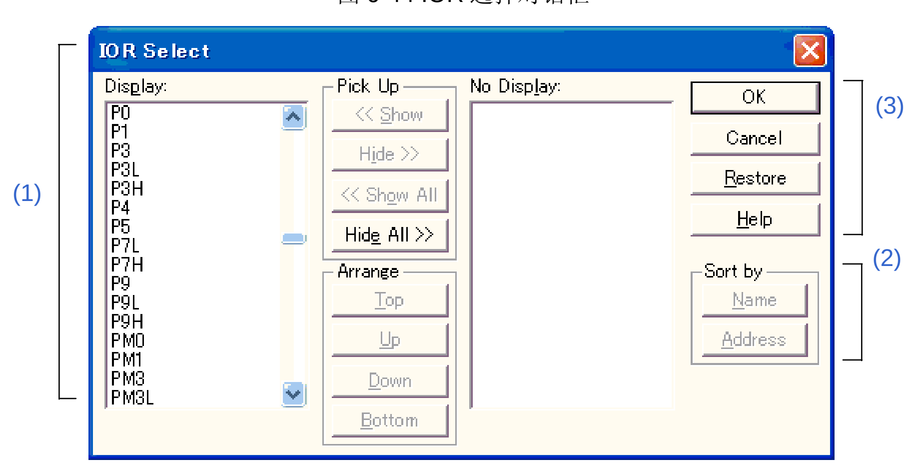
Move... 移动	打开地址移动对话框。
Add Watch...添加观察	打开添加观察对话框。
Add I/O Port...添加I/O端口	打开增加I/O端口对话框。
Bin	显示二进制数
Oct	显示八进制数
Dec	显示十进制数
Hex	显示十六进制数(缺省)
Sort By Name	按字母表顺序显示
Sort By Address	按地址顺序显示 (缺省)
Unsort不排序	不排序
Attribute属性	切换"属性"打开/关闭显示。
Pick Up提取	只显示IOR选择对话框中选择的寄存器。
Select...选择	打开IOR选择对话框。
Compulsion Read强制读取	强制读取那些为避免数值改变而禁止读取的周边I/O寄存器，或者在添加I/O端口对话框中添加的I/O端口和I/O保护区域的数据。

IOR 选择对话框

这个窗口用来选择没有在IOR窗口显示的周边I/O寄存器和I/O端口。(参考"5.8 寄存器操作功能".)

也可以用来指定寄存器和端口的显示顺序。

图 6-44 IOR 选择对话框



- 打开
- 每个区域的解释

打开

当IOR窗口是当前窗口时，选择[View]菜单中 [Select...]选项.

每个区域解释

(1)　显示周边I/O寄存器选/择显示顺序改变区域

区域	按钮	说明
Display: 显示		在IOR窗口显示的寄存器。 按住Ctrl或者Shift键并且单击上面按钮中的任一个可以选择两个和多个寄存器。
Pick Up提取		下面按钮用来改变要显示的外围I/O寄存器。
	<< Show	将选择的寄存器从 "No Display:" 列表移到"Display:"列表.
	Hide >>	将选择的寄存器从 " Display:" 列表移到" No Display:"列表.
	<< Show All	移动所有寄存器到"Display:".列表
	Hide All >>	移动所有寄存器到"No Display:"列表.
Arrange排列		下列按钮用来改变"Display:"中显示的顺序. 如果改变了显示顺序，不能选择多个行。一次只能选择一行。
	Top	移动选择的寄存器到表的最顶端
	Up	将选择的寄存器向上移动一行
	Down	将选择的寄存器向下移动一行
	Bottom	移动选择的寄存器到表的最底端
No Display:不显示		在IOR窗口中不显示的寄存器。

(2)　显示顺序改变按钮

区域	按钮	说明
Sort by排序法则		在"No Display:"中下列按钮用来改变显示的顺序.
	Name	按字母表顺序显示.
	Address	按地址顺序显示.

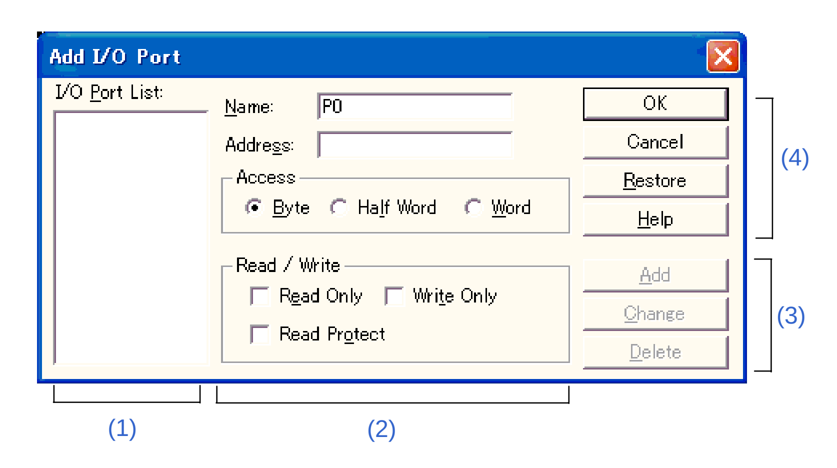
(3)　功能按钮

按钮	说明
OK确认	反显IOR窗口中这个对话框中选择的内容，关闭对话框。
Cancel取消	放弃修改并关闭对话框。
Restore恢复	恢复窗口打开之前的状态。
Help帮助	显示对话框在线帮助文件。

增加 I/O 端口对话框

这个对话框用来将I/O端口登录到IOR窗口中 (参考"5.8　寄存器操作功能".)

图 6-45：增加 I/O 端口对话框



- 打开
- 每个区域的解释

打开

选择 [Option] 菜单中 [Add I/O Port...]选项.

每个区域解释

(1)　I/O端口列表:

这个区域用来对当前登录的I/O端口列表显示

如果登录一个新的I/O端口，它将被增加到这个列表中。通过(3)按钮可以改变或删除一个已经登录的I/O端口

(2) I/O端口指定区域

Name:名称	这个区域用来指定添加的I/O端口的名称(最多15个字符长度).	
Address:地址	这个区域用来指定添加的I/O端口的地址 输入数据缺省为十六进制。也可以用符号指定地址(参考"表5-6 指定符号".) 这个区域中可以设置的地址可以是一个目标区域地址或者周边I/O寄存器地址。	
Access访问	这个区域用来选择添加的I/O端口的访问宽度。	
	Byte	8比特单位(缺省)
	Half Word	16比特单元
	Word	32比特单元
Read / Write读/写	这个区域用来指定添加的I/O端口的访问属性。 在缺省方式下，所有的属性都不选中 (例如I/O端口可以被读写)。	
	Read Only	只读
	Write Only	只写
	Read Protect	读保护

(3) 按钮

Add添加	添加一个指定地址的I/O端口。
Change修改	修改"I/O Port List:"中选择的I/O端口的设置。
Delete删除	删除"I/O Port List:"中选择的I/O端口的设置。

(4) 功能按钮

OK确认	在IOR窗口中反映添加的内容，关闭此对话框。
Cancel取消	放弃修改并关闭对话框。
Restore恢复	恢复最初状态。
Help帮助	显示对话框在线帮助文件

定时器对话框

[IECUBE]

这个对话框用来设置和登录定时器事件条件，同时显示执行时间的测量结果 (参考"5.12 事件功能" 和 "5.9 定时器功能 [IECUBE]".)

单击<View Always>按钮，"执行时间显示区域"可以作为定时器结果对话框一直显示。

通过在对话框中设置每个项目(最多256个)并单击<OK> 按钮，登录和设置定时器事件条件。使用事件管理器对登录的定时器事件条件进行管理。

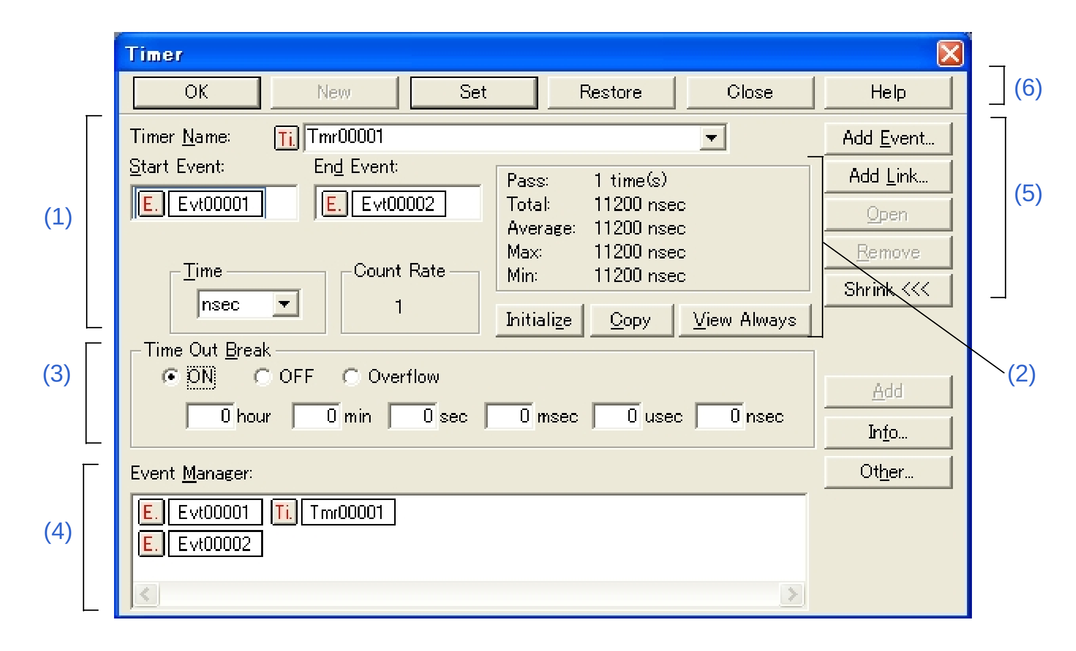
可同时使用的有效的定时器事件条件数目是有限的。(参考"5.12.4每个事件条件下可使能的事件数目".)

当选择定时器事件条件，将显示执行时间测量的结果

备注1: 即使在用户程序的执行期间，在RRM功能的每个采样时间可以实时更新测量结果。

备注2: 即使在用户程序执行期间，定时器事件条件设置/使能/禁止/删除操作都是允许的。

图 6-46 定时器对话框



- 打开
- 每个区域解释
- 注意事项

打开

单击**Tim按钮**或选择 [Event]菜单中的[Timer...]选项

每个区域的解释

(1) 定时器事件条件设置区域

Timer Name: 定时器名称	这个区域用来设置定时器事件的名称。 直接输入字符串作为事件名称，最多可以输入8个字符。 要显示已经创建的事件条件的内容，从下拉式菜单中选择。 要显示用户程序从执行到断点，指定"运行-断点". (参考"5.9.2 运行-断点事件".) 左边区域的标记指示了事件的使用状态。(参考"表5-18 事件图标".) 灰色标记指示事件条件正在被编辑，而且还没有登录。 单击左边的标记，可以使事件条件有效或者无效。
Start Event: 开始事件 End Event: 结束事件	这个区域用来为定时器设置定时器事件条件。 这个区域中可以登录的事件条件数目包括起始和结束条件。 可以通过简单的从事件管理区拖放事件图标到这个区域完成事件条件的设置。详情请参考"5.12.3设置事件条件"。
Time时间	此区域用来选中在(2)执行时间显示区域要显示的单元。
Count Rate计数速率	此区域显示定时器计数率值(在扩展选项对话框 (2) Timer [IECUBE] 中设置的值)。(当前的定时器事件条件显示时，显示在事件创建时的定时器计数率值。当新创建事件时，或者当Run-Break事件条件显示时，显示当前的定时器计数率值。)此区域不能被编辑。当前的定时器事件条件的定时器计数率值需要修改时，首先，修改扩展选项对话框中的(2) Timer [IECUBE]的值。然后，复位定时器事件条件。当事件创建时的定时器计数率值与在扩展选项对话框中(2) Timer [IECUBE]中设置的值不同时，超时时间被影响。因此此区域中的值显示为红色(参考 "注意事项")。

(2) 执行时间显示区域

这个区域显示程序执行时间的测量结果。

只有"Total" 在运行断点时显示。

当计数器溢出时，测量结果是不可信的，显示为红色。

Pass: 通过	通过的次数
Total: 总的	由起始事件和结束事件条件指定的测量区域中的总的执行时间。
Average: 平均	平均执行时间
Max: 最大	最大执行时间

Min: 最小	最小执行时间
<Initialize初始化>	清除测量结果。
<Copy复制>	用文本方式复制测量结果到剪贴板
<View Always总是显示>	打开定时器结果对话框。

表 6-14 可测量的数值

连接的IE	可测量的执行时间	可测量的执行次数
[IECUBE]	(总数，最大，最小 = 33bit， 外部 时钟 = 50 MHz) 最大大约 2.8 分钟. (1 division，分辨率 = 20 nsec) 最大最大大约 195.2 小时(4K division，分辨率= 81920 nsec)	16 bit 最大65,535 次.

(3) 超时断点

这个区域用来设置在"起始事件，终止事件"中为段测量时间指定的超时断点。(即从定时器事件开始建立到定时器结束事件之间的时间).

ON	如果段测量时间超过指定的超时时间，将发生一个超时断点事件(终止执行) 在文本框中指定超时时间。可以指定直到最大测量时间的数值。
OFF	没有超时断点发生(缺省)
Overflow溢出	如果段测量时间超过最大的测量时间，发生一个超时断点事件(终止执行) (参考"表 6-14 可测量值").

备注: 在这种运行事件条件下，这个区域固定为"OFF"位置，而且是禁止操作的

(4) 事件管理区域

这个区域用来显示登录的事件

在每个事件设置对话框中通过将此区域内显示的事件图标拖放到事件设置区域即可实现对事件条件的设置(参考"5.12.3 设置事件条件".)

这个区域对所有的事件关联对话框是共用的(参考"(2) 断点对话框中的事件管理区".)

(5) 功能按钮 (用来显示事件条件内容等)

这些按钮用来显示和删除事件条件设置区域中显示的事件条件，并且可以显示和隐藏事件管理区。这个区域对于所有的事件关联对话框是相同的。 (参考"断点对话框中的(3) 功能按钮 (用作事件条件显示，等。)")

(6)　功能按钮 (用来实现事件条件的登录、删除、使能和禁止)

这些按钮用来实现事件条件的登录、删除、使能和禁止。

通过单击<OK> (或<Set>)按钮来对指定事件条件的事件进行登录。这个区域在所有的事件关联对话框中是相同的 (参考”断点对话框”中"(4) 功能按钮(用来实现事件条件的登录、删除、使能和禁止")

注意事项

对于IECUBE，所有的定时器事件采用同一个定时器计数速率，因此，超时断点设置的时间随着事件条件中定时器计数速率的不同而改变(这个值在扩展选项对话框中 (2) Timer [IECUBE]中设置) 。

例:

在下面情况下：事件创建时定时器事件的定时计数率设为"16"并且超时时间设为"1 秒"。

(1)　当前定时器计数速率为"32"

->当前的定时器计数率值是事件创建时的值的一倍。这导致了2秒后一个定时器结束断点事件的发生。

(2)　当前定时器计数速率为"4"

->当前的定时器计数率值是事件创建时的值的1/4。这导致了250毫秒后一个定时器结束断点事件的发生。

因此，在上面情况下不要设置超时断点。(当超时断点被设为"OFF" 或者 "Overflow。"时，没有问题)

设置超时断点的方法：修改扩展选项对话框中(2) Timer [IECUBE]定时器计数率值，然后复位定时器事件。

定时器结果对话框

[IECUBE]

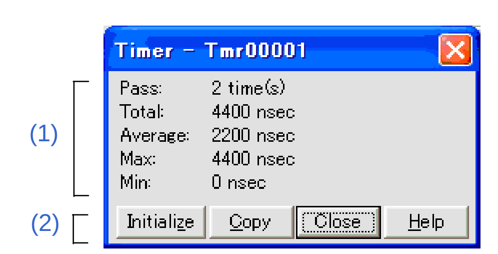
这个对话框显示称许执行时间的测量结果 (参考"5.9 定时器功能 [IECUBE]".)

单击定时器对话框中<View Always> 按钮，相应每个定时器时间条件的对话框被一对一打开。两个和多个对话框可以同时打开。

可以同时打开最多256 + 1 (运行-断点事件)个定时器结果对话框，可以同时测量的事件数目是"5.12.4 每个事件条件可使能的事件数目" + 1 (运行断点事件) 个有效事件。

备注: 即使在用户程序的执行期间，每次RRM功能的采样时将对测量结果进行更新。

图 6-47：定时器结果对话框



- 打开
- 每个区域的解释

打开

单击<View Always> 按钮选择定时器对话框中定时器事件条件。

每个区域的解释

(1) 执行时间显示区域

这个区域和定时器对话框相同。

Pass:通过	执行的次数
Total:总和	在指定起始点和事件条件之间执行的总时间
Average: 平均	平均执行时间
Max: 最大	最大执行时间
Min: 最小	最小执行时间

(2) 功能按钮

Initialize初始化	清除所有测量结果
Copy复制	将测量结果以文本方式复制到剪贴板上
Close关闭	关闭对话框
Help帮助	显示对话框在线帮助文件

跟踪查看对话框

[IECUBE]

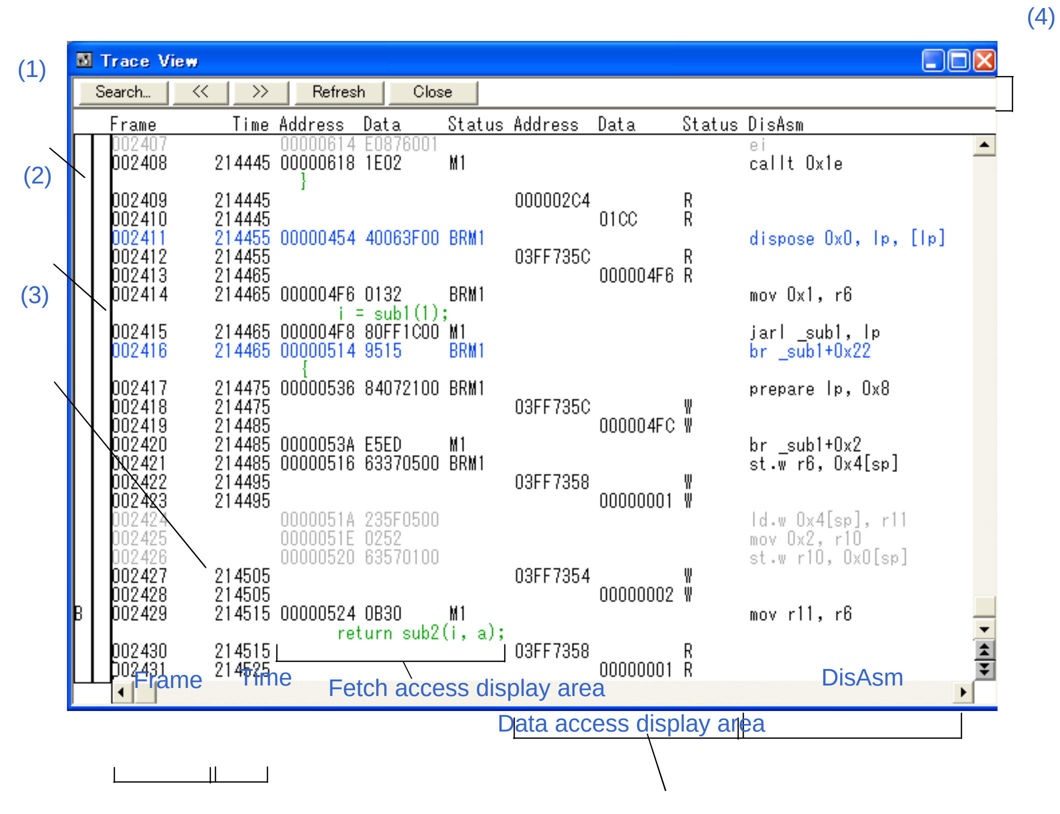
这个窗口用来显示跟踪结果(参考"5.10 跟踪功能[IECUBE]".)

在断点和单步执行期间更新显示结果。

这个窗口具有混和显示模式(跟踪观察窗口)。同时还有"5.16.3 跟踪结果关联窗口[IECUBE]".

通过上下文菜单、功能按钮，等等可以实现其它的操作。

图 6-48：跟踪查看对话框



- 打开
- 每个区域解释
- [View]菜单 (跟踪观察窗口专用功能)
- 上下文菜单

打开

单击TrW按钮，或选择[Browse]中的[Trace]选项.

每个区域解释

(1)　点标记显示区域

这个区域用来显示事件设置状态 (事件标记)。

如果在相应的跟踪地址设置了执行事件或访问事件，将显示与事件类型相对应的标记。

这个标记不是跟踪期间的标记，而是跟踪结果显示时的设置的标记。

(2)　跟踪模式显示区域

这个区域显示跟踪模式的类型

A	起始或者结束frame (段跟踪或者限定跟踪)
T	延时触发frame
M	DMA点访问frame (DMA起始点和结束点) (参考"5.10.7 DMA 点跟踪功能".)
N	部分存取跟踪数据frame

(3)　跟踪结果显示区域

这个区域显示跟踪的结果。

补充帧用灰色显示。

补充帧中，只显示"Frame", "Fetch access display area", 和"DisAsm" 。

无论下列的子项是否被显示，都可以通过跟踪数据选择对话框对其进行选择。

Frame帧	这个区域用来指定帧的数目。
Time时间	此区域显示以下情况下所需时钟计数：目标芯片从程序开始执行到每个帧的一条指令开始执行或者存储器访问的发生。 在补充帧中，此区域不显示。 显示内容在数据选择对话框中可以在时钟计数显示和时间显示之间切换。 对于定时计数分频比和最大测量时间之间的关系，参考"表6-5 定时标志计数器分频比和最大测量时间(定时标志计数(跟踪))之间的关系"。使用外部时钟(50MHz)执行时间标志和时间的测量。 **注：**如果发生溢出，时间标志最大值显示为红色。
Fetch access display area 取访问显示区域	此区域显示取指结果。 对于补充帧，只显示"地址"和"数据"。

项目	子项目	状态	说明
	Address地址		显示取指的地址
	Data数据		显示取指的数据
	Status状态		有以下状态类型
		BRM1	取跳转后的第一条指令的第一个字节(此帧包括BRM1和M1。) 如果取地址是符号的起始，第一行高亮显示为蓝色。
		M1	取指令的第一个字节。
		Blank	下面任一项 - 数据访问帧(最多3帧。(当RRM功能被选中最多2 帧)) - 第二条指令的帧(当两条指令同时执行时) -无效帧
		IF	加入中断并且被取消的取指地址(在此地址处断点被设置)
		INFO	开始跟踪(延时触发跟踪)，跟踪开始/跟踪结束(节跟踪)，取指地址(限定跟踪开始/结束)
Data access display area 数据访问显示区域			此区域显示数据访问结果。 在补充帧中，此区域不显示。
	Address地址		显示访问地址
	Data数据		显示访问数据
	Status状态		显示访问状态
		R	数据读
		W	数据写
DisAsm反汇编			这个区域显示反汇编结果 (仅当"Status"为 BRM1, M1, 或补充帧). 对于只显示访问地址的帧，代之显示的是全局符号.

备注1: 当一个 6个字节或 8个字节指令码被显示的时候，最初4个字节显示在第一个帧，而其他字节在第二个和第三个帧显示。 如果二个指令同时运行，一个帧在两行被显示。较低地址的指令码在第一行显示，较高地址的指令码在第二行显示。

备注2: 在两个指令同时执行时，为了显示指令代码，第一条指令在第一行显示，另一条在第二行显示。

(4) 功能按钮

Search... 搜索	打开跟踪搜索对话框并且搜索跟踪结果。搜索结果在跟踪观察窗口中高亮显示。[View] -> [Search...]执行同样的功能
<<	前向搜索(在屏幕中向上)满足在跟踪搜索对话框中设置的搜索条件的跟踪结果。
>>	后向搜索(在屏幕中向下)满足在跟踪搜索对话框中设置的搜索条件的跟踪结果。
Refresh刷新	以最新数据更新当前窗口内容。
Close关闭	关闭对话框。

[View] 菜单 (跟踪观察窗口专用功能)

当观察窗口是激活状态时，下列的内容可以增加到 [View] 菜单中。

Select... 选择	选择要显示的内容。 打开跟踪数据选择对话框。
Mix混合	切换是否以混合显示模式显示源文件。 选中：混合显示 未选中：隐藏(缺省)
Window Synchronize窗口同步	和下面的窗口一起连接到跟踪查看窗口：(参考"5.16.3 关联窗口的跟踪结果 [IECUBE]".) 连接到一个选中的窗口。
Source Text源文件文本	连接到源文件窗口。
Assemble汇编	连接到汇编文件窗口。
Memory存储	连接到存储器窗口。

上下文菜单

Move... 移动	移动显示位置。 打开跟踪移动对话框。
Trace Clear跟踪清除	清除跟踪数据。
Select... 选择	选择显示的数据。 打开跟踪数据选择对话框。
Mix混合	切换是否以混合显示模式显示源文件。 选中：混合显示 未选中：隐藏(缺省)
Window Synchronize窗口同步	和下面的窗口一起连接到跟踪观察窗口(参考"5.16.3 关联窗口的跟踪结果 [IECUBE]".)
Source Text源文件文本	连接到源文件窗口。
Assemble汇编	连接到汇编窗口。
Memory存储	连接到存储器窗口。

Source Text源文件文本	使用光标处的数据值作为跳转目的地址，显示相应的原文件文本和源代码行。(参考 "5.16.2 跳转功能"。)但是，如果在跳转目的地址处不存在行信息，将不能跳转。 打开源文件窗口。 如果源文件窗口以激活状态打开，该窗口显示在最前面(以便对其进行操作)。
Assemble汇编	从以光标位置处数据值指定作为跳转目的地址反汇编并且显示。(参考 "5.16.2跳转功能") 打开汇编窗口。 如果源文件窗口以激活状态打开，该窗口显示在最前面(以便对其进行操作)。
Memory存储	把光标处的数据值指定的跳转目的地址作为起始地址，显示相应的存储内容。(参考 "5.16.2 跳转功能"。)打开存储窗口。 如果源文件窗口以激活状态打开，该窗口显示在最前面(以便对其进行操作)。

跟踪搜索对话框

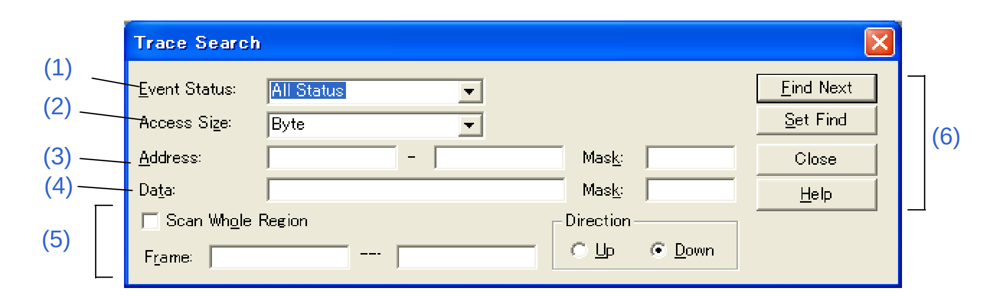
[IECUBE]

这个对话框用来在跟踪观察对话框中进行搜索。(参考"5.10 跟踪功能[IECUBE]".)

设置每一项，并且单击 <Find Next> 按钮，开始搜索。

单击<Set Find> 按钮，在跟踪观察窗口中的方向按钮 (<< 和>>) 可用来实现搜索。

图 6-49：跟踪搜索对话框



- 打开
- 每个区域解释

打开

当跟踪查看窗口是当前窗口时，选择[View] 菜单中[Search...]按钮或单击该窗口中的<Search... >按钮。

每个区域解释

(1)　事件状态:

这个区域用来选择状态条件。

如果一个状态条件被省略，将搜索所有的帧(所有状态)。

备注: 不搜索IF和INFO帧。

All 状态	所有帧(缺省)
M1取指	M1 取指　(包括 BRM1取指)
R/W	数据读/写(包括队，写)
Read读	读数据
Write写	写数据

(2)　访问宽度:

这个区域用来设置访问宽度条件。

通过选择一个访问宽度条件，可以确定要检测数据条件的访问宽度

Byte字节	按8比特宽度搜索数据条件(只在8比特访问期间执行).
Half Word半字	按16比特宽度搜索数据条件(只在16比特访问期间执行)
Word字	按32比特宽度搜索数据条件(只在32比特访问期间执行).
No Condition没有条件	不进行基于访问宽度的搜索 ("Data"区域不能输入任何数据).
Bit比特	按1比特宽度搜索数据条件(只在8比特访问期间执行)**注1,2**

注1:　如果访问事件被指定为一个状态条件，可选择的比特不被显示。如果指定为比特或者1，将发生一个错误。

注2:　这种情况下，按照1比特宽度的数据条件进行搜索。由于仿真器的操作，不能直接检测到某一比特的访问；仿真器按照下面的内部设置地址条件和数据条件搜索一个哑元：

输入 举例:		跟踪搜索的设置:
Address: FE20.1	->	Address: FE20
Data: 1		Data: 00000010B
		Mask: 11111101B

如果同一地址的另外一比特被访问，或者具有同一地址所有的8比特都被访问，那么，如果地址和比特与指定值相匹配[地址.bit]，根据指定状态检测一个事件。

备注:　如果没有指定访问宽度条件，系统根据数据条件和地址条件作出自动判断，地址条件设置如下：

- 比特，如果地址条件设置为比特单位
- 字节，如果地址条件设置为8比特单位
- 半字，如果地址条件设置为16比特单位
- 字，如果地址条件设置为32比特单位
- 无条件，如果没有指定条件

(3) 地址条件设置区域

这个区域用来指定地址条件 (可被省略)。 可设置的内容如下：

表 6-15 地址条件设置区域(跟踪)

可设置范围	条件
0 <= address value <= 0xFFFFFFFF	None
0 <= mask value <= 0xFFFFFFFF	None

<table>
<tr><td rowspan="4">Address:地址</td><td colspan="2">设置一个地址条件(低地址-高地址)(可以忽略)。输入数值缺省为十六进制。也可以用一个符号或者表达式指定一个符号。(参考"表5-6 指定符号"。)
可设置项如下：</td></tr>
<tr><td>Setting as a point作为断点设置</td><td>设置低地址的值为唯一，或者低地址和高地址设为同一值。</td></tr>
<tr><td>Setting as a range 作为范围设置</td><td>设置低地址的值为唯一，或者低地址和高地址设为同一值。</td></tr>
<tr><td>Setting as a bit作为比特设置</td><td>设置低地址的值为唯一，或者低地址和高地址设为同一值。
以"address.bit"格式指定一个值。不能设置屏蔽。指示比特位置的比特值必须符合以下条件：0<=比特值<=7。</td></tr>
<tr><td>Mask:屏蔽</td><td colspan="2">设定一个地址值的屏蔽值(只有当"作为一个点设置")(可能被忽略)。屏蔽值为1的比特位的地址值可能为0或者1。</td></tr>
</table>

例1: 0x4000 to 0x40FF 满足此条件。

Address:地址	0x4000 to 0x4000
Mask:屏蔽值	0xFF

例2: 0x4000, 0x4001, 0x4100, and 0x4101满足此条件。

Address: 地址	0x4000 to 0x4000
Mask: 屏蔽值	0x101

(4) 数据条件设置区域

这个区域用来设置数据条件(可被省略)。

根据"(2) Access Size"中指定的访问宽度条件，确定可设置区域的范围(参考"事件对话框" 中"(5) 数据条件设置区域".)

Data:数据	设置一个数据值作为数据条件。输入数值缺省为十六进制。也可以使用一个符号来指定数据。(参考 "表5-6指定符号")
Mask:屏蔽值	为数据值设置一个屏蔽值(可以忽略)。 当设置一个屏蔽后，屏蔽值为1的比特数据值可能为0或1。

例1: 0x4000 to 0x40FF满足此条件。

Data数据	0x4000
Mask屏蔽	0xFF

例2: 0x4000, 0x4001, 0x4100, and 0x4101满足此条件。

Data数据	0x4000
Mask屏蔽	0x101

(5) 搜索条件设置区域

<table>
<tr><td>Scan Whole Region扫描整个区域</td><td colspan="2">.选中以搜索整个指定区域</td></tr>
<tr><td>Frame:</td><td colspan="2">此区域用来指定要搜索的一个frame序号。
输入数值的缺省基数为十六进制。符号也可以由frame序号指定格式来指定。</td></tr>
<tr><td rowspan="3">Direction方向</td><td colspan="2">这个区域用来指示搜索的方向</td></tr>
<tr><td>Up</td><td>前向搜索。从当前光标位置开始向前搜索数据(屏幕中向上)。</td></tr>
<tr><td>Down</td><td>后向搜索。从当前光标位置开始向后搜索数据(屏幕中向下)。</td></tr>
</table>

(6) 功能按钮

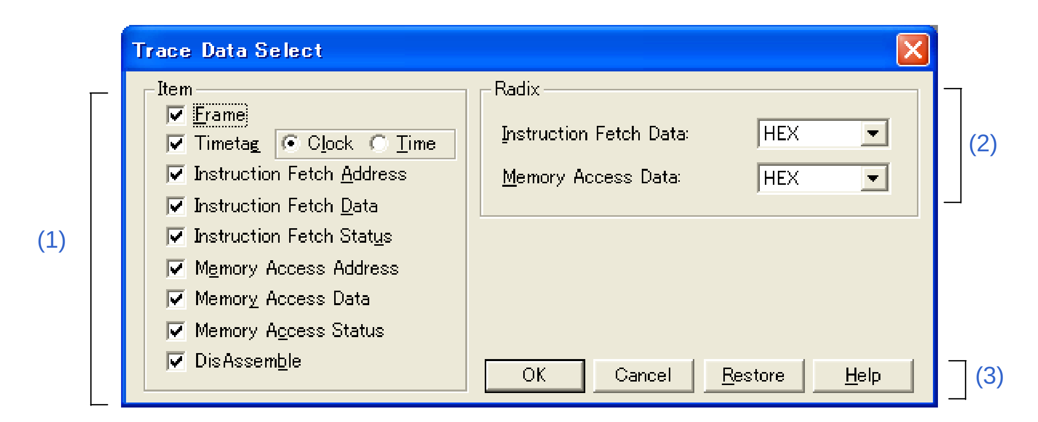
Find Next查找下一个	根据给定条件搜索指定数据。 如果搜索到指定的帧，将高亮显示。要继续搜索，再次点击此按钮。
Set Find设置搜索	设置指定条件作为搜索条件并且关闭对话框。
Close关闭	关闭对话框
Help帮助	显示对话框在线帮助文件。

跟踪数据选择对话框

[IECUBE]

这个对话框用来选择在跟踪查看对话框中显示的条目 (参考"5.10 跟踪功能[IECUBE]".)

图 6-50：跟踪数据选择对话框



- 打开
- 每个区域解释

打开

当跟踪查看窗口是当前窗口时，选择 [View] 菜单中的 [Select...] 选项。

每个区域解释

(1) 项目

这个区域用来选择在跟踪观察窗口中显示的条目。下列显示区域可能或不能被选择，将显示选择的区域(参考"(3) 跟踪结果显示区域".)

Frame帧	帧域
Timetag	时间域 选择显示"Clock" 或者 "Time"
Instruction Fetch Address取地址指令	地址(取指访问显示区域) 域
Instruction Fetch Data取数据指令	数据(取指访问显示区域) 域
Instruction Fetch Status取状态指令	状态(取指访问显示区域) 域
Memory Access Address存储区访问地址	地址(数据访问显示区域) 域
Memory Access Data存储区访问数据	数据(数据访问显示区域) 域
Memory Access Status存储区访问状态	状态(数据访问显示区域) 域
DisAssemble反汇编	DisAsm域

(2)　数制

这个区域用来选择显示数据的数制。显示下列选中或没选中的项目。

Instruction Fetch Data:取数据指令	数据(取指访问显示区域) 域
Memory Access Data:存储访问数据	数据(数据访问显示区域) 域

HEX	显示十六进制数(缺省)
DEC	显示十进制数
OCT	显示八进制数
Bin	显示二进制数

(3)　功能按钮

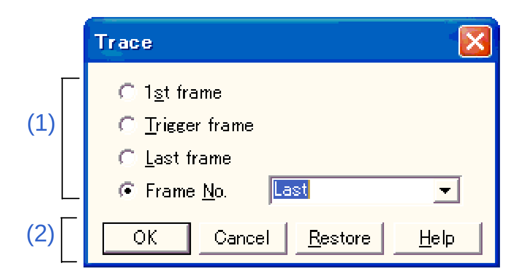
OK确认	在跟踪观察窗口中的对话框反映选中的内容。
Cancel取消	关闭对话框
Restore恢复	恢复初始状态
Help帮助	显示对话框在线帮助文件

跟踪移动对话框

[IECUBE]

这个对话框用来指定跟踪观察对话框显示的起始位置（参考"5.10　跟踪功能　[IECUBE]".)

图 6-51：跟踪移动对话框



- 打开
- 每个区域解释

打开

当跟踪查看窗口是当前窗口时，选择[View]菜单中 [Move...]选项.

每个区域解释

(1)　帧选择区域

这个区域用来指定目的帧

1st frame	移动显示的起始位置到跟踪数据的第一个帧。
Trigger frame触发frame	移动显示的起始位置到跟踪数据的触发帧中。
Last frame最后frame	移动显示的起始位置到跟踪数据的最后一个帧。
Frame No. frame编号	移动显示的起始位置到指定的帧(参考"表6-16 frame号指定格式".) 在缺省条件下， 调用此对话框或者"Last"的窗口中选择的字符串被选中。 输入数据缺省为十进制。 如果指定了0，显示的起始位置移动到跟踪数据的第一个帧。最多可以保留16个输入的历史记录

表 6-16 帧号指定格式

描述	缩写	内容
+numeric value数字值	None	从光标处的帧显示起始位置向后移动(屏幕向下)指定的帧数(数字值)。
-numeric value数字值	None	从光标处的帧显示起始位置向前移动(屏幕向下)指定的帧数(数字值)。
Top顶部	O	移动显示的起始位置到第一个帧。
First第一个	S	和"1st frame"相同
Trigger触发	T	和"Trigger frame"相同
Last最后一个	L	和"Last frame"相同

(2)　功能按钮

OK确认	从指定位置开始跟踪显示。
Cancel取消	关闭对话框。
Restore恢复	恢复输入数据到初始的状态。
Help帮助	显示对话框在线帮助文件。

跟踪对话框

[IECUBE]

这个对话框用来登录、设置和显示跟踪事件条件。 (参考"5.12 事件功能", "5.10 跟踪功能[IECUBE]".)

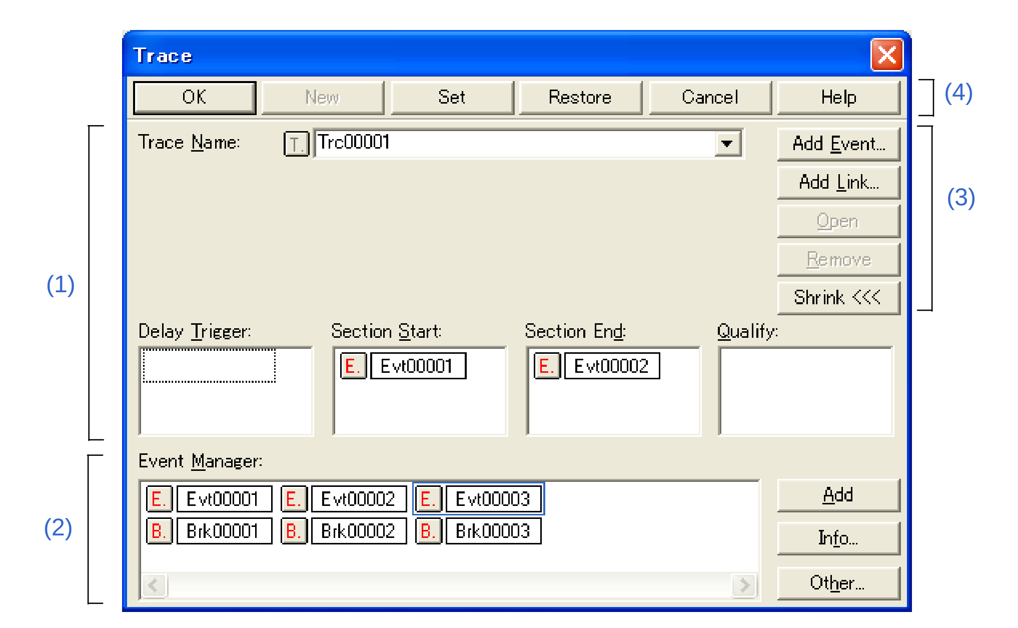
在这个对话框中可以设定执行条件跟踪时的跟踪事件条件。 (参考"表5-13 条件跟踪类型".)

通过在这个对话框中设置每个项目(最多256项)并单击<OK>按钮，实现对跟踪事件条件的登录和设置。通过事件管理器对登录的跟踪事件条件进行管理。

可以被同时使用(有效)的跟踪事件条件的数目是有限的。 (参考"5.12.4 每个事件条件可使能的事件数目".)

备注: 即使在用户程序执行期间，跟踪事件条件可以被设置/使能/禁止和删除。在这种情况下，在处理期间跟踪器的操作回立刻停止。

图 6-52：跟踪对话框



- 打开
- 每个区域解释

打开

单击**Trc按钮**或选择[Event] 菜单中[Trace...]功能.

每区域解释

(1)　跟踪事件条件设置区域

Trace Name:跟踪名	此区域用来设置一个跟踪事件名字。直接输入一个最多8个字符的字符串作为一个名字。 从下拉列表选择显示一个已创建的事件条件内容。此区域左边的标志只是事件的使用状态。(参考"表 5-18 事件图标 "。) 灰色标志说明正在编辑事件条件，并且尚未登录。单击左边的标志，可以使事件条件生效或者无效。
Delay Trigger:延时触发	此区域用来为一个延时触发设置事件条件。(参考"5.10.6设置条件跟踪") 可以如下容易的设置事件条件：把要设置的事件图标从事件管理器中拖放在到这个区域。详情请参考"5.12.3设置事件条件"。此区域中可以设置的事件条件个数参考"表6-17可设置的事件个数" 。
Section Start: 起始区 Section End:	可以如下容易的设置事件条件：把要设置的事件图标从事件管理器中拖放在到这个区域。详情请参考"5.12.3设置事件条件"。此区域中可以设置的事件条件个数参考"表6-17可设置的事件个数" 。
Qualify:限定	此区域用来设置限定跟踪的事件条件。(参考 "5.10.6设置条件跟踪") 如果设置了两个以上事件，每个事件发生时进行跟踪。 可以如下容易的设置事件条件：把要设置的事件图标从事件管理器中拖放在到这个区域。详情请参考"5.12.3设置事件条件"。此区域中可以设置的事件条件个数参考"表6-17可设置的事件个数" 。

表 6-17：可设置事件的数目

连接的 IE	事件条件 总共 (执行/访问)	事件链接条件
[IECUBE]	14 (8[*a]/6)	1

***a**　　执行后可用的事件

(2) 事件管理器区

这个区域用来显示登录事件的列表。

通过把这个区域中显示的事件图标放置到每个事件设置对话框中的事件设置区域可以容易设置每个事件条件。(参考"5.12.3 保存事件条件"。)

这个区域对所有事件关联对话框是相同的。 (参考" 断点对话框"中的"(2) 事件管理区")

(3) 功能按钮 (事件条件内容显示，等等。)

这些按钮用来显示或者删除事件条件设置区域中显示的事件条件，以及显示或隐藏事件管理器。这个区域对所有事件关联对话框是相同的。 (参考"断点对话框"中的"(3) 功能按钮 (事件条件内容显示等")

(4) 功能按钮(登录，删除，使能和禁止事件条件)

这些按钮用来登录，删除，使能和禁止事件。

通过单击<OK> (或<Set>) 按钮可以对指定事件条件的事件进行登录和设置，这个区域对所有事件-关联对话框是相同的。 (参考"(4) 断点对话框中的功能 按 钮 (登录，删除，使能和禁止事件条件).)

延时计数对话框

[IECUBE]

这个对话框用来设置和显示延时计数值（参考"5.10 跟踪功能[IECUBE]"。）

通过设置延时计数的数值，当跟踪对话框中设定的延时触发事件条件满足时，一个跟踪过程可以根据延时计数设置的数值多次执行(参考"5.10.6 设置条件跟踪".)

图 6-53：延时计数对话框



- 打开
- 每个区域解释

打开

选择[Event] 菜单中的[Delay Count...]功能.

每个区域的解释

(1)延时计数设置区域

Delay Count延时计数	可以选择以下项目。	
	FIRST	把触发指针放在第一个跟踪数据上，跟踪所有的帧，然后停止跟踪。
	MIDDLE	把触发指针放在跟踪数据的中部，跟踪所有的帧的一半，然后停止跟踪。
	LAST	把触发指针放在最后一个跟踪数据上，然后立即停止跟踪。

注意事项:　　根据扩展选项对话框中设置的跟踪存储区大小不同，延时计数的值可能不同。

例：当跟踪存储区大小是256K帧时，

FIRST: 256K - 1 frame

MIDDLE: (256K / 2) - 1　frame

LAST: 5　frames

(2)　功能按钮

OK确认	使设置生效并关闭对话框。
Restore恢复	恢复先前的设置。
Cancel取消	关闭对话框。
Help帮助	显示这个对话框在线帮助文件。

代码覆盖率窗口

[IECUBE]

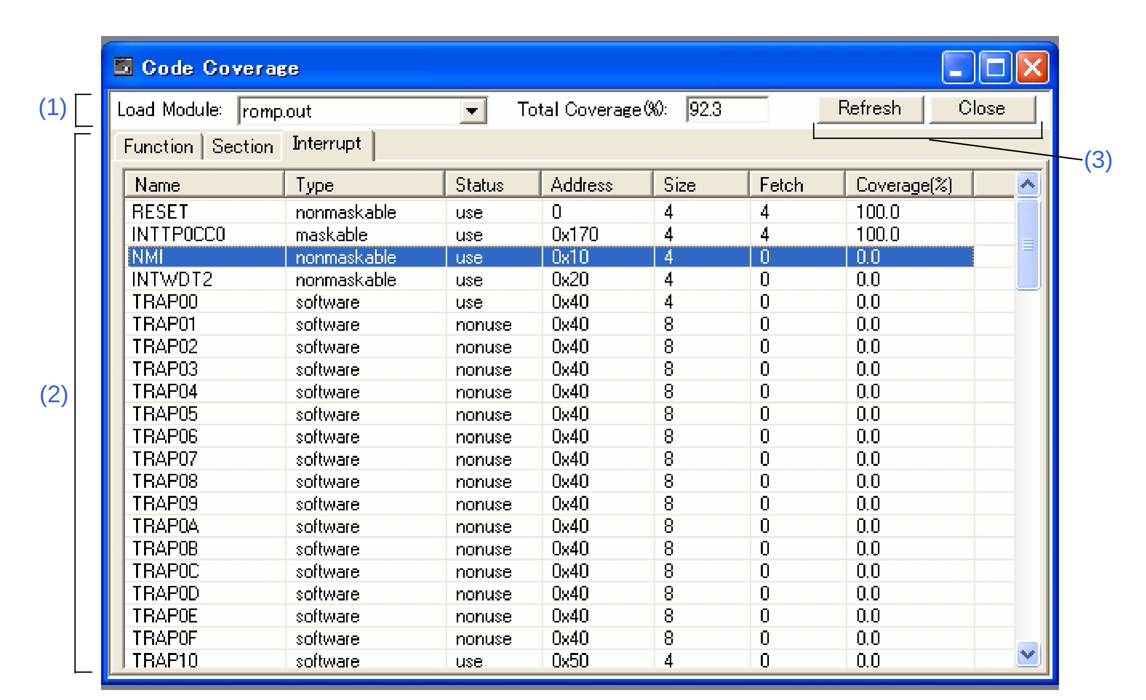
这个对话框用来显示代码覆盖率测量的结果 (C0覆盖率). (参考 "5.11　覆盖测量功能[IECUBE]".)

可以在源文件窗口或者汇编窗口中检查用户程序行已经执行或者没有执行。

注意事项:　如果片上的闪存数据通过仿真器自编程程序被替换，覆盖率测量结果是不准确的。(参考" 闪存选项对话框".)

备注:　当ID850QB起动时，覆盖率数据被清除。

图 6-54：代码覆盖窗口



- 打开
- 每个区域解释
- 上下文菜单

打开

单击**Cov** 按钮，或选择 [Browse] 菜单的 [Code Coverage]选项.

每个区域解释

(1) 覆盖信息显示区

Load Module:加载模块	这个区域用来选择已经下载的加载模块文件。 如果没有下载加载文件，此区域是空的。
Total Coverage (%):总的覆盖率	这个区域用来显示已经测量过代码覆盖率的区域的覆盖率值。 **总的覆盖率＝总的执行函数大小 (存取) /总的函数大小** (不包括覆盖测量范围外的段) 当没有下载加载模块文件时，这个区域是空的

(2) 测量结果显示区

每个制表符区域用来显示不同的测量结果(函数，段和中断句柄)。

在发生断点时，覆盖率测量结果是自动更新的 (在用户程序执行期间不更新)。

当没有下载加载模块文件时，这个区域是空的。

使用选中的行的起始地址值作为一个跳转指针，显示从当前制表符跳转到源文件窗口或者汇编窗口。跳转的目的窗口从这个跳转指针开始显示。

通过选择源跳转行然后选择[Jump]菜单中的 [Source Text/Assemble] 可以实现跳转功能。通过双击跳转源行也可以实现跳转功能。

备注: 点击每个列的标题(在标签上)可以实现显示内容的排序 (标号) (每次单击一次标题实现升序和降序的切换).

(a) 当选择[Function] tab时

Name名称	函数名称 (在汇编源文件情况下，以文件函数为单位显示)
File文件	定义函数的文件名称
Address地址	函数的起始地址
Size大小	函数大小 (单位：字节)
Fetch取指	执行的字节数目 (取指)
Coverage (%)覆盖率	函数的覆盖率(0 - 100%) ---- : 当函数超出覆盖率测量的范围

(b) 当选择[Section] tab时

Name名称	段名称
Type类型	段类型 (代码，数据)
Address地址	段起始地址
Size大小	段大小 (单位：字节)

Fetch取指	执行的字节数目 (存取)
Coverage [%]覆盖率	覆盖率的段 (0 - 100%) ---- : 当段位于覆盖率测量区域之外时

(c)　当选择 [Interrupt] tab时

Name名称	中断请求名称
Type类型	中断类型 (非屏蔽，屏蔽，软件，安全ID，闪存屏蔽选项)
Status状态	使用程序状态 ---- : 未知
Address地址	中断句柄的起始地址。
Size大小	中断句柄的大小 (单位：字节) 除了"use"之外最大状态值
Fetch取指	执行的字节数 (取指)
Coverage [%]覆盖率	中断句柄的覆盖率(0 - 100%) ---- : 当中断句柄超出覆盖率测量的范围

(3)　功能按钮

Refresh刷新	更新当前窗口内容为最后数据。
Close关闭	关闭窗口

上下文菜单

Source Text源文本	使用光标位置的数据值作为跳转目的地址，显示相应的源文件文本和源文件行。(参考"5.16.2 跳跃功能".) 但是，如果在跳转目的地址处不存在行信息，不可以跳转。 打开源文件窗口。如果打开激活的源文件窗口，这个窗口在前面显示(因此可以对其进行操作)。
Assemble汇编	从光标处的数据值所指定的跳转目的地址处开始反汇编并且显示。(参考"5.16.2 跳跃功能".) 打开汇编文件窗口。如果打开激活的汇编文件窗口，这个窗口在前面显示(因此可以对其进行操作).
Clear清除	清除覆盖率测量的结果
Select... 选择	选择覆盖率测量的范围，1 MB或者更多。 打开覆盖率-地址对话框

覆盖-地址对话框

[IECUBE]

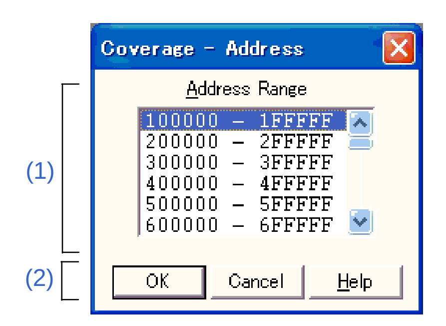
这个窗口用来选择代码覆盖率窗口中显示的覆盖率测量范围 。(参考"5.11 覆盖率测量功能 [IECUBE]".)

可测量的范围如下所示。在这个对话框中，任何1 MB的范围都可被选择。

表 6-18：覆盖率测量范围 (详细)

连接的IE	代码覆盖率测量范围
[IECUBE]	- 1 MB 地址空间 0x000000 to 0x0FFFFF(固定的测量区域) - 从0x100000 到 0x3FFFFFF任何1 MB地址空间 (由此对话框可选择) (缺省: 从0x3F00000到 0x3FFFFFF)

图 6-55：覆盖-地址对话框



- 打开
- 每个区域解释

打开

选择 [Option] 菜单中[Coverage] 中的 [Select...]选项

每个区域解释

(1) 地址选择区域

Address Range地址范围	此区域用来选择1 MB空间以进行覆盖率测量。改变测量范围会清除以前选中区域的测量结果(覆盖率数据)，但是不要清除从0 to 0x0FFFFF固定测量区域中的覆盖率数据。要决定覆盖率测量范围，选中此区域，然后点击<OK>按钮。

(2)　功能按钮

OK	使选择的覆盖率测量范围有效。
取消	关闭对话框
帮助	显示对话框在线帮助文件

软件断点管理器

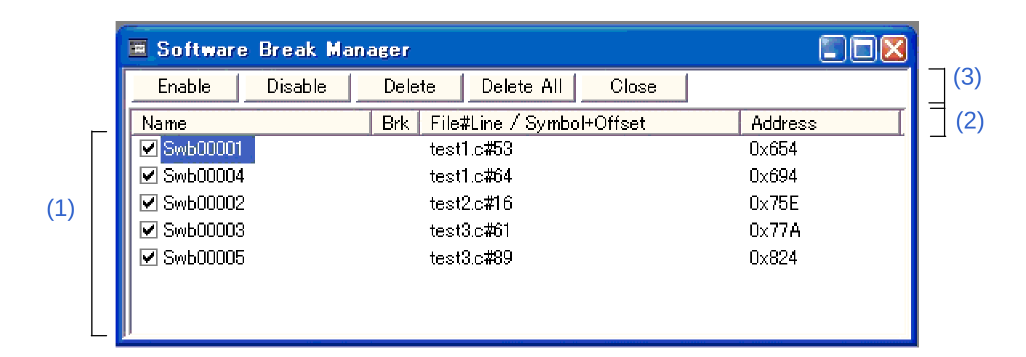
这个窗口用来显示、使能、禁止和删除软件断点。(参考"5.4.4 硬件断点和软件断点".)
在这个窗口中不能设置软件断点，软件断点在源文件窗口和汇编窗口中设置。(参考"5.4.2 断点设置"。)

注意事项1: 在用户程序执行期间，软件断点可以被设置和删除的。在用户程序执行期间，软件断点可以被设置、删除、使能和禁止。每次使用用户程序产生断点都会显示一个告警信息。

注意事项2: 当伪实时监视功能(读出时断点)为使能状态时，不能设置任何软件断点。而且如果设置一个合法的软件断点，用户程序执行期间的写操作(DMM功能)是禁止的。

备注: 在每一列的题目上单击可以实现显示条目按顺序排列 (每单击一次标题，升序/降序转换一次)。

图 6-56：软件断点管理器



- 打开
- 每个区域的解释

打开

选择[Event] 菜单中的 [Software Break Manager]选项.

每个区域解释

(1) 断点信息显示区域

Name名字	这一个区域显示登录事件的名字，以及指示每个事件是否允许或者禁止。 缺省方式下，以 "Swb+[数目]" 的形式显示一个事件名字。 它能被修改到一个最多256个字符的字符串。 要改变事件名字， 选择和单击一个名字。 然后直接编辑名字。 为了设定编辑，按回车键。 当一个事件被使能的时候，选中检查框。 要禁止一个事件，不选择检查框。 此外，双击源行相对应的事件名字，名字跳跃到源文件窗口，然而如果它与源行不相对应，名字跳到汇编窗口。
Brk	">"标志指明在当前PC位置处设置的软件断点事件(以便容易地识别导致断点的软件断点事件)
File#Line / Symbol+Offset 文件#行 / 符号+偏移量	此区域显示如下面所设置软件断点事件产生的位置： **Program$file name#line number** (如果事件与源行相对应。) **Program$file name#symbol+offset** (如果事件与源行不相对应。) 当符号重新下载时，事件以此为基础进行估计。
Address地址	此区域显示设置软件断点事件的地址。

(2) 项目标号区域

通过单击每个项目标号可对显示的项目顺序排序 (每单击一次标题，升序/降序转换一次)。

Name名称	根据字母表顺序排序(升序/降序)
Brk	不排序
File#Line / Symbol+Offset 文件#行 / 符号+偏移量	根据字符串顺序排序 (升序/降序)
Address地址	根据地址的高低排序 (升序/降序)

(3) 功能按钮

Enable使能	使能所有选择的事件
Disable禁止	禁止所有选择的事件
Delete删除	删除所有所有选择的事件
Delete All全部删除	删除所有设置的软件断点事件。
Close关闭	关闭窗口。

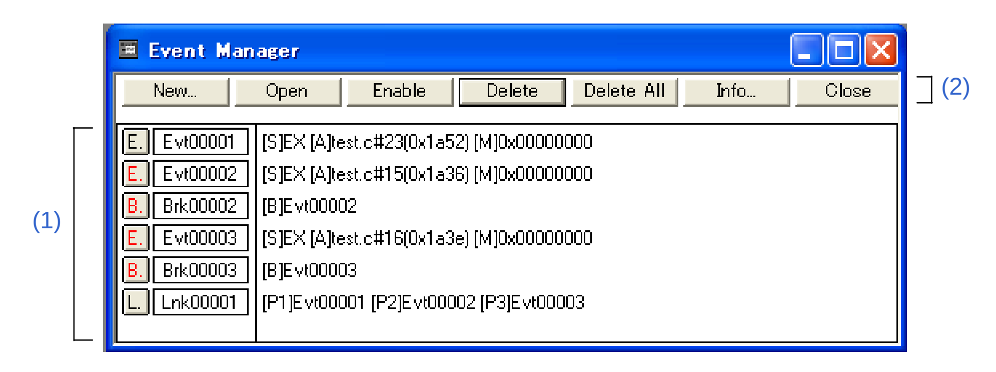
事件管理器

这个窗口用来管理事件条件，这个窗口允许显示，允许/禁止，删除多种事件条件 (参考"5.12 事件功能".)

在这个窗口中可以进行上下文菜单和功能按钮的其他操作。

事件图标是跳转功能中的跳跃指针。

图 6-57：事件管理器(详细显示方式)



- 打开
- 每个区域解释
- [View] 菜单(事件管理器专用功能)
- 上下文菜单

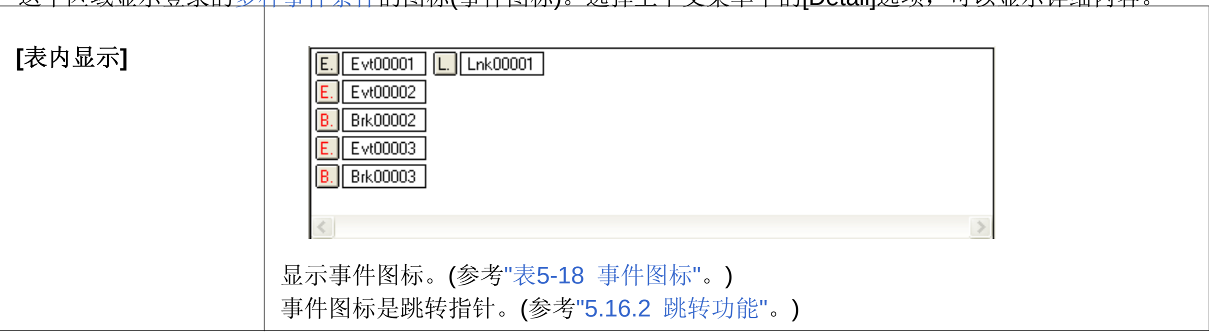
打开

单击**Mgr按钮**或选择 [Event] 菜单中 [Event Manager]选项.

每个区域解释

(1)　事件显示区域

这个区域显示登录的多种事件条件的图标(事件图标)。选择上下文菜单中的[Detail]选项，可以显示详细内容。



[表内显示]	显示事件图标。(参考"表5-18 事件图标"。) 事件图标是跳转指针。(参考"5.16.2 跳转功能"。)

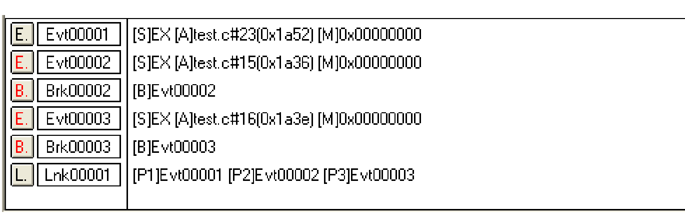
[In detailed display]	 使用下面的关键信息作为分隔符，显示事件内容的细节 。 (参考 "表6-19显示事件细节的分隔符")

表 6-19：显示事件细节的分隔符

关键信息	内容
事件条件	
[S]	状态条件
[Z]	访问大小条件
[A]	地址条件 符号或者表达式(实际地址)
[D]	数据条件 符号或者表达式：(实际地址)
[M]	屏蔽条件
事件连结条件	
[P1] - [P4]	第n行的事件关联条件
[D]	禁止条件
[P]	执行次数计数条件
断点条件	
[B]	断点条件
跟踪条件 [IECUBE]	
[M]	跟踪控制模式
[T]	延时触发条件
[D]	延时计数
[S]	跟踪开始条件
[E]	跟踪结束条件
[Q]	限定跟踪条件
定时器条件 [IECUBE]	
[S]	定时器测量起始条件
[E]	定时器测量结束条件
[U]	定时器测量单位
[B]	超时断点条件

(2)　功能按钮

OK	自动登录编辑的事件条件，如果有的话，并且关闭此对话框。 每个事件条件登录后立即生效。
New...	打开[New Event]对话框。 New Event: Event... / Event Link... / Break... / Trace... / Timer... / Cancel 打开对话框创建一个新的事件条件。单击每个按钮，相应的事件设置对话框可以使用新的事件名字设置。事件设置对话框打开后，此对话框被关闭。单击<Cancel>按钮返回事件管理器。
Set 设置	登录多种事件条件。因为即使事件已登录，对话框也不关闭，可以登录新事件条件。每个事件条件登录后立即生效。
Open 打开	根据相应的选中的事件条件(一个)，打开多种事件设置对话框。每个设置对话框显示选中的事件条件的内容。双击事件图标或者按下回车键执行同样的操作。
Enable 使能 Disable 禁止	使有效(使能)或者使无效(禁止)选中的事件条件。但是，事件条件和事件关联条件，不能被使能或者禁止。单击事件图标标志执行同样的操作。
Remove 删除	删除选中的事件。当要删除一个事件条件或者一个事件关联条件时，产生一个错误，并且如果事件被用作多种事件条件，事件条件或者事件关联条件不能被删除。
Delete All 全部删除	删除所有的事件条件。
Info...信息	打开[Event Info]对话框。此对话框用来修改显示模式并且重排列事件名字。 Event In...: Sort by Name / Sort by Kind / Unsort / Detail / Overview / Cancel <Sort by Name> .....根据名称排序。 <Sort by Kind> .... 根据事件类型排序。 <Unsort> .... 显示已经登录但没有排序的事件。 <Detail> .... 设置详细的显示模式。 <Overview> .... 设置显示模式。 <Cancel> ....关闭此对话框(和 ESC 键功能相同)。
Close 关闭	关闭此对话框

[View] 菜单(事件管理器-专用功能)

当事件管理器处于激活状态时，下列条目可以增加到 [View] 菜单中。

Select All Event选择所有事件	选择所有登录的事件。
Delete Event删除事件	删除一个选择的事件。
Sort By Name按名字排序	按事件名称的顺序显示图标。
Sort By Kind按种类排序	按事件类型的顺序显示图标。
Unsort不排序	图标不排序。(缺省)
Detail细节	显示细节。
Overview总揽	列表显示(缺省)

上下文菜单

Sort By Name按名称排序	按事件名称的顺序显示图标。
Sort By Kind按种类排序	按事件类型的顺序显示图标。
Unsort不排序	图标不排序。(缺省)
Detail细节	显示细节。
Overview总揽	列表显示(缺省)
Source Text源文本	显示相应的源文本和源行，使用选中的事件作为跳转目的地址。(参考"5.16.2跳转功能") 但是，如果在跳转目的地址处不存在行信息，无法跳转。 打开源文件窗口 If an active Source Window is open, that window is displayed in the forefront (so that it can be manipulated).如果一个源文件窗口以激活状态打开，该窗口显示在最前面(以便操作)。
Assemble汇编	从选中的事件位置显示汇编窗口，此位置用作跳转目的地址。(参考"5.16.2跳转功能"。) 打开汇编窗口。 如果汇编窗口以激活状态打开，该窗口显示在最前面(以便操作)。
Memory存储	从选中的事件位置显示存储区内容，此位置用作跳转目的地址。(参考"5.16.2跳转功能"。) 打开存储窗口。 如果存储窗口以激活状态打开，该窗口显示在最前面(以便操作)。

事件对话框

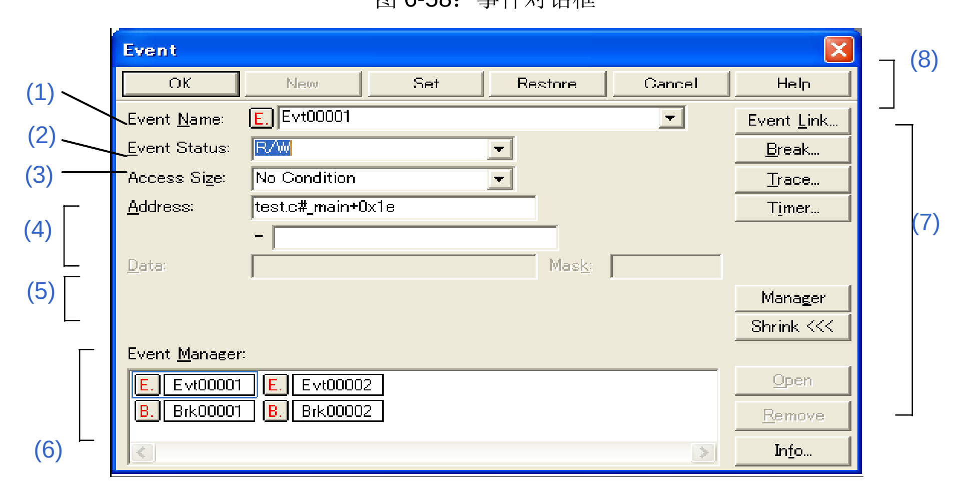
这个对话框用来登录和显示事件条件。(参考"5.12 事件功能".)

在对话框中设置每一项，单击<OK>按钮可以实现对事件条件的设置。

使用事件管理器可以管理登录的事件条件。

为多种事件条件可以设置一个事件条件。然而，可以同时使用的事件条件数目是有限的。(参考"5.12.4每个事件条件下可使能的事件数目"。)

图 6-58：事件对话框



- 打开
- 每个区域解释

打开

正常模式

如果采用下面的方式打开事件对话框，可以登录不指定目的的事件。


单击**Evn**按钮，或选择 [Event]菜单中 [Event...]选项。

选择模式

当事件对话框处于如下打开状态，如果按下<OK> 按钮，事件条件可以在这个设置对话框中登录 (打开此对话框的设置对话框显示在标题栏。)

在每个不同的事件设置对话框，单击<Add Event... > 按钮。

每个区域解释

(1)　事件名称:

这个区域用来设置事件名称。

直接输入文字字符串作为事件名称，最多8个字符。

为了显示已经创建的事件条件的内容，可以从下拉菜单中选择。

在这种选择模式下，在事件对话框中事件条件设置区，可以对选择的事件条件进行设置。

左边区域标记指示了事件的使用状态。(参考"表5-18 事件图标"。) 灰色的"E"表示正在编辑的事件条件还没有登录。

(2)　事件状态:

可以被指定的状态条件列表如下。

这个区域用来选择状态条件。

通过指定状态条件，执行事件的类型和访问事件是确定的 (如果指定一个执行事件，任何内容不能加入在"Access Size:" 和"Data:, Mask:"项目中)。

执行事件		
Execution执行	EX	执行程序
Before Execution执行前	EX-B	执行程序 (执行之前断点) 注1
访问事件		
R/W	RW	数据读/写
Read读	R	数据读
Write写	W	数据写
R/W(Data not Equal)数据不相等	RWND	数据读/写 (如果数据条件不满足，发生一个事件错误)注2
Read(Data not Equal) 数据不相等	RND	数据读 (如果数据条件不满足，发生一个事件错误)注2
Write(Data not Equal) 数据不相等	WND	数据写 (只有数据条件不满足时，发生一个事件错误)注2

注1:　多个条目可被指定，但是只有两个条目可被使能，包括访问事件。
这个地址范围不能被指定。只能使用断点事件条件。

注2:　在这种情况下，数据条件不能省略
对于访问大小条件，"No Condition" 和"Bit" 不能被选择

(3)　访问空间:

这个区域用来选择访问大小条件。

通过从下拉菜单选择访问大小条件，确定一个访问事件就可以检测数据条件的访问宽度。

如果没有指定访问的大小，由地址条件和数据条件可以自动地作出判断。

Byte字节	在8比特宽度条件下检测数据条件 (只在8比特存取期间执行).
Half Word半字	在16比特宽度条件下检测数据条件 (只在16比特存取期间执行).
Word字	在32比特宽度条件下检测数据条件 (只在32比特存取期间执行).
No Condition无条件	不检测访问的大小(数据区域不能输入数据).
Bit比特	在1比特宽度条件下检测数据条件 (只在8比特访问期间执行)[注].

注: 在这种情况下，进行一比特宽度的数据搜索。因为是仿真器操作，访问一个比特不能被直接检测到，仿真器根据下面内部设置的地址条件和数据条件搜索一个哑比特：

输入例子:		跟踪搜索的设置:
Address: 3FF7000.1	->	Address: 3FF7000
Data: 00000010B		Mask: 11111101B

如果同一个地址的另一个比特被访问或者同一个地址的8个比特都被访问，根据指定的地址和比特与指定值匹配情况检测相应的事件[address.bit]。如果地址和比特与指定值匹配[address.bit]，根据指定状态检测事件。

(4)　地址条件设置区域

这个区域用来指定地址条件(可被省略)。

下列内容可以设置。

表 6-20：可设置的地址条件的范围 (事件)

连接的 IE	可设置区域
[IECUBE] **[MINICUBE]**	0 <= address value <= 0xFFFFFFFF

Address: 地址	设置地址条件(低地址-高地址)(可能被忽略)。输入数值的数制为十六进制。也可以使用符号或者表达式指定符号。(参考"表5-6指定符号"。) 下面可以设置：	
	Setting as a point	只为低地址设置一个值，或者把低地址和高地址设为相同值。
	Setting as a range	只为低地址设置一个值，或者把低地址和高地址设为相同值。
	Setting as a bit	只为低地址设置一个值，或者把低地址和高地址设为相同值。 以"address.bit"格式指定一个值。屏蔽不能设置。指示比特位置的比特值，必须符合下面条件：0 <= bit<=7

注意事项:　指定28比特地址，因为物理地址和映射空间在事件设置中是可以区分的。

(5)　数据条件设置区域

这个区域用来指定一个数据条件(数据值，屏蔽值)。

缺省的数值输入数制是十六进制。

根据"Access Size:"中指定的访问大小条件，可设置区域根据下面而不同。

Data:数据	此区域用来指定一个数据条件(数据值，屏蔽值)缺省的输入数字数制为十六进制。 设置区域如下变化，依赖于在"Access Size:"中指定的访问大小条件。	
	Byte字节	0 <=数据值<= 0xFF 0 <=屏蔽值<= 0xFF
	Half Word半字	0 <=数据值<= 0xFFFF 0 <=屏蔽值<= 0xFFFF
	Word字	0 <=数据值<= 0xFFFFFFFF 0 <=屏蔽值<= 0xFFFFFFFF
	Bit位	数据值 = 0 or 1 屏蔽值 =不能被指定。
Mask:屏蔽	为数据值设置一个屏蔽值(可以被忽略)。当屏蔽值设定以后，屏蔽值为1的数据值为0或者1。	

例1: 0x4000 to 0x40FF 满足此条件。

Data数据	0x4000
Mask屏蔽	0xFF

例2: 0x4000, 0x4001, 0x4100, and 0x4101 满足此条件。

Data数据	0x4000
Mask屏蔽	0x101

(6) 事件管理区域

Event manager:事件管理器	这个区域用来显示登录事件的列表(参考"表5-18 事件图标", "(4) 事件管理区的操作".)
<Open>打开	打开选中的事件条件(1个)相应的多个事件设置对话框。每个设置对话框显示选中的事件条件的内容。双击事件图标或者按下回车键，执行同样的操作。
<Remove>删除	删除选中的事件。 当删除事件条件或者事件关联条件的时候，产生一个错误，如果事件用作一个多种事件条件时，事件条件或者事件关联条件不能被删除。
<Info...>信息	打开[Event Info]对话框。 此对话框用来改变显示模式和重排事件名字。 <Sort by Name> .....根据事件名称进行排序 <Sort by Kind> .... 根据事件类型进行排序 <Unsort> .... 显示那些已经登录但没有排序的事件 <Detail> .... 设置详细的事件显示模式 <Overview> .... 设置列表显示模式 <Cancel> .... 关闭对话框 (和ESC键功能相同).

(7) 功能按钮 (事件条件内容显示，等等)

Event Link... Break... Trace... Timer...	单击每个按钮，可以用新事件名字设置相应的事件设置对话框。
Manager...	打开事件管理器。
Expand >>> Shrink <<<	打开或者关闭事件管理器显示区域。 增加或者减小对话框大小。

(8) 功能按钮 (登录，删除，使能和禁止事件条件)

这些按钮用来实现事件的登录、删除、使能和禁止功能。

通过单击<OK> (或<Set>) 按钮用来对指定的事件条件进行登录或设定。这个区域对所有的事件-关联对话框是通用的。 (参考在"断点对话框""中(4) 功能按钮(用来实现对事件条件的登录、删除、使能和禁止)".)

事件关联对话框

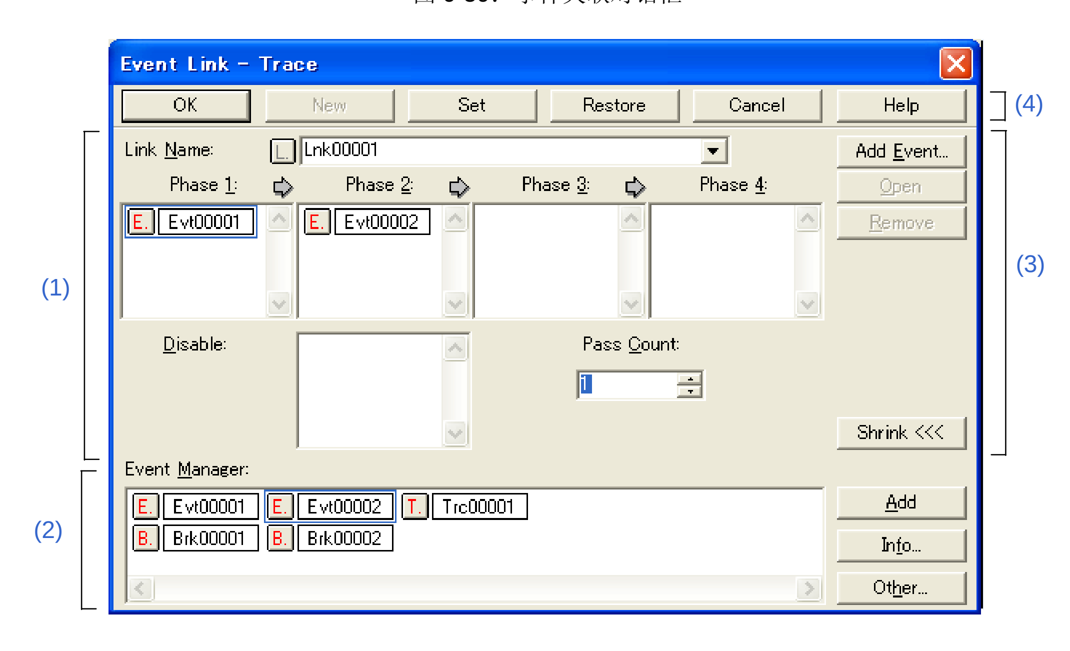
这个对话框用来登录和显示事件关联条件（参考"5.12 事件功能"。）

在这个对话框中设置每个项目(最大256项)，然后单击<OK>按键，可以进行事件关联条件的登录。

使用事件管理器管理登录的事件关联条件。

但是，可以同时使用的事件关联条件数目是有限的(参考"5.12.4 每个事件条件下可使能的事件数目")。

图 6-59：事件关联对话框



- 打开
- 每个区域解释

打开

正常模式

如果按照下面的方法打开事件关联对话框，可以在不指定事件目的的条件下登录事件关联条件。

选择[Event] 菜单中[Event Link...]选项。

选择模式

如果事件关联对话框按下面的方式打开并按下<OK>按钮，事件关联条件可以登录在这个对话框打开的设置对话框中。

在每个多种事件设置对话框中单击<Add Link... >按钮。

(在标题栏上显示设置对话框，事件关联对话框从中打开。)

每个区域解释

(1) 事件关联条件设置区域

Link Name:	直接输入长度最多8个字符的字符串作为一个名字。 要显示一个早已创建的事件关联条件的内容，从下拉列表选择。 在选择模式下，选中的事件条件可以在设置对话框的事件关联条件设置区域中进行设置，该设置对话框调用事件关联对话框。 此区域左面的标志指示事件关联条件的使用状态(“表5-18事件图标")。灰色的"L"标志指示正在编辑的事件关联条件仍未被登录。
Phase 1: Phase 2: Phase 3: Phase 4:	此区域用来指定事件条件和检测事件的顺序。 最多可指定4个顺序。 但是如果程序执行期间检测到一个被禁止的条件，到目前为止被满足的事件条件被初始化，从第一个事件条件开始，事件条件再次被检测。 如果同时检测到一个被禁止的条件和一个关联条件，被禁止的条件优先。 设置 "Phase 1" -> "Phase 2" -> "Phase 3" -> "Phase 4", 在这种次序下， "Phase 4" 并不需要设置。这种情况下，当为最后的phase设置的事件条件被检测到时，产生一个事件。 事件条件只可以为"Phase 1"设置，或者同样的事件条件可以为两个以上Phases进行设置。 可以被设置到此区域的每个phase和对话框的事件条件的个数，参考”表6-21 事件关联对话框中的事件条件个数”。 可以如下容易的设置事件条件：从事件管理器区域拖动要设置的事件图标并放进这个区域。详情，请参考”5.12.3设置事件条件”。
Disable:	此区域用来设置一个事件条件，用来使到目前为止被满足的事件条件无效。 可以被设置到此区域的每个phase和对话框的事件条件的个数，参考”表6-21 事件关联对话框中的事件条件个数”。 可以如下容易的设置事件条件：从事件管理器区域拖动要设置的事件图标并放进这个区域。详情，请参考”5.12.3设置事件条件”。
Pass Count: **[IECUBE]**	此区域用来设置一个通过计数条件。 通过计数条件指定程序运行中在给定条件被满足之前一个事件条件必须被满足多少次 如果没有指定通过计数条件，假定为1(事件条件被满足后此条件例及被满足)

表 6-21：事件关联对话框中事件条件的数目

连接的 IE		每个区域	总数(执行/访问)
[IECUBE]		10	14 (8/6)
[MINICUBE]	Nx85ET（RCU0+TEU+TRCU）	10*a	12 (8/4)
	Nx85E901 （RCU0)， RCU1	1*b	2*c
[MINICUBE2]	带调试功能	1*b	2*c
	不带调试功能	-	-

***a** 可以在阶段1-4设置 (在执行前)

***b** 只能在阶段1-2设置 (必须在两个阶段设置)

***c** 仅执行前断点

(2) 事件管理器区

这个区域用来显示登录的事件。

通过把此区域中的事件图标拖放至事件设置对话框中的事件设置区域可以容易地实现事件条件的设置。 (参考 "5.12.3 设置事件条件".)

这个区域在所有的事件关联对话框中是相同的(参考" 断点对话框"中的"(2) 事件管理器区域".)

(3) 功能按钮(事件条件内容显示等)

这些按钮用来显示和删除事件条件设置区域中的事件条件，显示和隐藏事件管理器。这个区域在所有的事件关联对话框中是相同的。 (参考"断点对话框中(3) 功能按钮 (用来显示事件条件)" in the " Break Dialog Box".)

(4) 功能按钮(登录，删除，使能和禁止事件条件)

这些按钮用来实现事件的登录、删除、使能和禁止。通过单击<OK> (或<Set>)按钮可以实现事件条件的设置

OK确认	自动登录正在编辑的事件条件并关闭对话框。 **在选择模式下** 选中一个事件条件并再次显示调用事件关联对话框的设置对话框 (在标题条显示)。如果调用对话框已经关闭，选择模式回至正常模式并关闭事件对话框。否则此对话框将关闭。
New新建	在此对话框中创建一个新的事件条件。 新建一个事件条件并自动生成事件名。
Set设置	登录不同事件条件。因为即使登录事件后对话框也不关闭，所以可以登录新的事件条件。 **在选择模式下** 选中一个事件条件。如果正在编辑一个事件，将自动登录并选择此事件。

Enable/Disable使能/禁止	使能或禁止选择的事件条件。 但是，事件条件和事件关联条件不能被使能或禁止。 单击时间图表标记将执行相同的操作。
Clear清除	清除事件条件的内容。
Restore恢复	恢复一个已编辑的事件条件的内容。 如果显示的是未登录事件条件，除事件名外的所有区域为空或者设置为缺省值。
Cancel取消 Close关闭	不登录事件条件并关闭对话框. 即使正在编辑这个事件条件。
Help帮助	显示对话框在线帮助文件

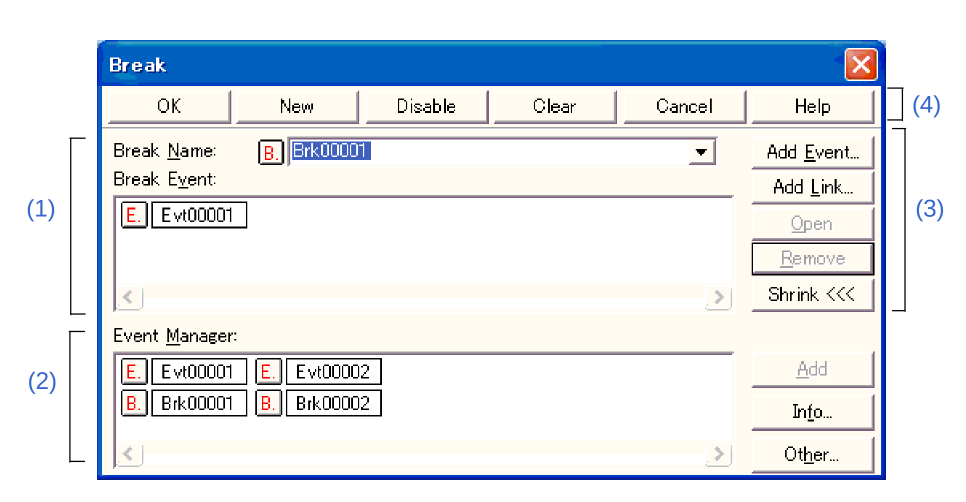
断点对话框

这个窗口用来登录，设置和显示断点事件条件 (参考"5.12 事件功能", "5.4 断点功能".)

通过在这个对话框中对每一项(最多256项)进行设置并单击<OK>按钮，可以实现断点事件条件的登录和设置。

在事件管理器中可以对登录和设置的断点事件条件进行管理。可以被同时设置(使能)的断点事件条件是严格受限的 (参考"5.12.4 每个事件条件下可使能的事件数目".)

图 6-60：断点对话框



- 打开
- 每个区域的解释

打开

 单击**Brk** 按钮或选择[Event]菜单中 [Break...]选项.

每个区域解释

(1) 断点事件条件设置区域

Break Name:断电名称	此区域用来设置断点事件名称。 输入最多八个字母或数字作为名称。 从下拉列表中选择显示已经创建的事件条件。 区域左边的标记表明事件的使用状态。 (参考 "表 5-18 事件图标"。) 灰色标记表明事件条件正在编辑，尚未登录。单击左面的标记使事件条件变为有效或无效。
Break Event:断点事件	这个区域用来设定断点事件条件。 这个区域可以设置事件条件的数目和事件关联条件的数目参考"表6-22 条件设置区域可设置事件的数目"。 可以通过把事件管理器中的事件图标拖放至此区域轻松实现事件条件的设置。 关于细节参考"5.12.3 设置事件条件".

表 6-22： 条件设置区中可设置事件的数目

Connected IE 连接的IE		总共的事件条件(执行前/执行后/访问)	事件关联条件
[IECUBE]		16 (2 / 8 / 6)	1
[MINICUBE]	Nx85ET （RCU0+TEU+TRCU）	14 (2 / 8 / 4)	-
	Nx85E901 （RCU0)， RCU1	2	-
[MINICUBE2]	带调试功能	2	-
	不带调试功能	-	-

(2) 事件管理区

这个区域用来显示登录的事件列表.

通过把此区域中的事件图标拖放至事件设置对话框中的事件设置区域可以容易地实现事件条件的设置。 (参考 "5.12.3 设置事件条件".)

这个区域在所有事件关联对话框中是相同的。

<Add>添加	将事件管理器中选择的事件条件和事件关联条件添加至焦点设置区域

<Info...>信息	打开[Event Info]对话框 这个对话框用来修改显示模式和重排事件名称
	Event In... Sort by Name Sort by Kind Unsort Detail Overview Cancel
	<Sort by Name>....... 按名称顺序对事件排序 <Sort by Kind> ....按类型顺序对事件排序 <Unsort> .... 按登录顺序显示事件而不排序 <Detail>.................... 设置显示模式的细节 <Overview>............. 设置列表显示模式 <Cancel> .... 关闭对话框(和ESC将功能相同).
<Other...>其它	打开[Set Other] 对话框
	Set Other Event... Event Link... Break... Trace... Timer... Manager Cancel
	单击每个按钮，将以新的事件名设置打开相应的事件设置对话框。 事件设置对话框打开后，这个对话框关闭 <Manager...> .... 打开事件管理器 <Cancel>................. 关闭创建事件条件的对话框

(3)　功能按钮 (为事件条件内容显示，等)

这些按钮用来显示和删除事件条件设置区域的事件条件，同时显示和隐藏事件管理器区。这个区域在所有其他的事件关联对话框中是相同的。

Add Event...添加事件	以选择模式打开事件对话框，选择或者新建一个欲设置的事件条件。 当 按下<Add Event...>按钮，事件条件将添加至选择的区域。
Add Link...添加链接	以选择模式打开事件关联对话框，和选择或者新建一个事件关联条件。 当 按下<Add Link...> 按钮，事件条件将添加至选择的区域。

Open打开	打开与选择的事件条件匹配的不同的事件设置对话框(一). 每个设置对话框显示选择事件条件的内容。双击事件图符或按回车键执行同样的操作。
Remove去除 Delete删除	删除选择的事件 当一个事件条件或一个事件关联条件将被删除时，如果事件被用作不同的事件条件，将发生一个差误，而且事件条件或者事件链接条件不能被删除
Expand >>> 扩展 Shrink <<< 收缩	打开或者关闭事件管理器的显示区域。 对话框的大小可以增加或者缩小。

(4)　功能按钮(登录、删除有效或无效的事件条件)

这些按钮用来登录、删除事件，使之有效或无效。

通过单击<OK>或<Set>按钮对有效的事件条件进行登录。这个区域在所有事件关联对话框中是相同的。

OK确认	自动登录事件条件，或者编辑不同的事件条件，同时关闭对话框。 一旦登录了事件条件，每个事件条件都是有效的。
New...新建	在对话框种信创建一个事件条件 事件条件名称自动产生，新的事件条件准备好。
Set设置	登录事件条件或各种不同的事件条件。 因为对话框甚至不被关闭，一个事件已经被登录,新的事件条件被登录。 每个事件条件变成有效的一旦它已经被登录。

Enable/Disable使能/禁止	使能或者禁止选择的事件条件。 但是，事件条件和事件关联条件不能被使能和禁止 当单击事件图标时执行同样的操作
Clear清除	清除事件条件的内容.
Restore恢复	恢复编辑的事件条件的内容 如果显示未登录的事件条件，除事件名外其他区域为空，或者设置为缺省值。
Cancel取消 Close关闭	关闭对话框。即使正在编辑一个事件条件，这个事件条件仍不能被登录，对话框关闭。
Help帮助	显示对话框在线帮助文件

查看文件保存对话框

这个对话框用来将当前窗口中显示的信息保存至一个查看文件（参考"5.15.2 窗口显示信息(查看文件)"。）

图 6-61：查看文件保存对话框

- 打开
- 每个区域的解释

打开

当被保存的窗口时当前窗口时选择[File] 菜单中[Save As...]选项.

每个区域解释

(1) 保存文件设置区域

Save in:保存至	这个区域用来指定文件名称。一个文件名可以直接输入，或者从这个区域的列表中选择。 扩展名最多可以指定257个字符的字符串。
File name:文件名	
Save as type:以某类型文件保存	这个区域用来指定保存文件的类型(扩展名)(参考 "表5-21查看文件类型".) 将显示与当前窗口扩展名相匹配的文件。

(2)　设置文件保存区域

如果指定的范围是100行/100帧/256字节或以上，一个消息对话框将指示保存的进度。如果中途需要停止保存，单击消息对话框中的<Stop> 按钮。

<table>
<tr><td rowspan="7">Save range包村范围</td><td colspan="2">指定保存数据的范围
如果当前窗口按下面的方式保存则显示这个区域：
-源文件窗口
-汇编窗口
-存储器窗口
-跟踪观察窗口</td></tr>
<tr><td>All所有</td><td>选择保存从第一行到最后一行的所有内容。</td></tr>
<tr><td>Screen shot抓屏</td><td>这种方式选择保存屏幕上的可见部分，即从屏幕的起始行到最后一行。如果源文件窗口是混合显示模式，将保存从屏幕上可见的源文件行开始的内容。</td></tr>
<tr><td rowspan="4">Specify Line
Specify Frame
Specify Address
指定行
指定帧
指定地址</td><td>选择保存区间的起始行和终止行
如果起始行和终止行被省略，第一行和最后一行被假设为起始行和结束行。
显示当前窗口的相应内容</td></tr>
<tr><td>指定行
指定保存行号的范围。
缺省输入数值为十进制
如果源文件窗口是混合显示模式，指定行的混合显示部分也将保存</td></tr>
<tr><td>指定结构 [IECUBE]
指定保存跟踪帧的范围 (参考"表6-16 指定帧数目格式".)　缺省输入数据是十进制。</td></tr>
<tr><td>指定地址
指定保存地址的范围
可以用符号和表达式指定地址 (Refer to "表5-6 指定符号".)
缺省输入数值为十六进制</td></tr>
</table>

(3)　功能按钮

Save保存	保存当前窗口显示信息到一个选择的文件。保存后关闭对话框。
Cancel取消	放弃修改并关闭对话框
Help帮助	显示对话框在线帮助文件
Restore恢复	对话框打开前恢复以前的设置

观察文件加载对话框

这个对话框用来读取观察文件(参考"5.15.2 窗口显示信息(观察窗口)".)

当加载一个查看文件，将打开一个参考窗口 (静态方式下的源文件窗口) 并显示上次保存时的显示信息。

根据下列加载的文件不同，打开的窗口及其状态是不同的。

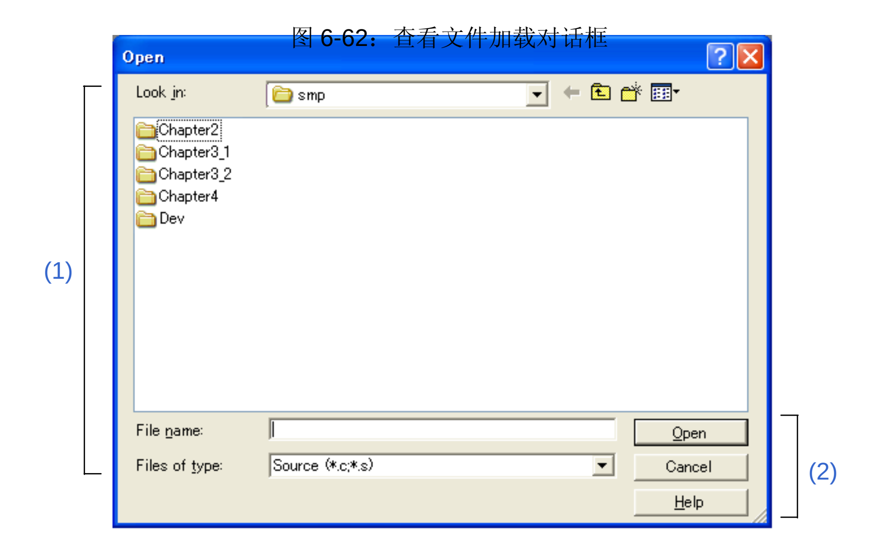
- **加载已经读入符号信息的源文件**

  如果有一个激活状态的源文件窗口，它以静态方式打开。否则，源文件窗口以激活方式打开。

- **加载没有读加载符号信息的源文件或查看文件**

  在源文件窗口中以静态方式打开一个文本格式窗口。

图 6-62：查看文件加载对话框



- 打开
- 每个区域解释

打开

单击Open按钮或选择 [File] 菜单中 [Open...]选项.

每个区域解释

(1)　加载文件设置区域

<table>
<tr><td>Look in:在某范围内查找</td><td rowspan="2">这个区域用来指定加载文件名称。一个文件名可以从键盘直接输入，或者从这个区域的列表中选择。
扩展名最多可以指定257个字符的字符串。</td></tr>
<tr><td>File name:文件名</td></tr>
<tr><td>Files of type:文件类型</td><td>这个区域用来指定加载文件的类型(扩展名) (参考"表5-21查看文件类型".)</td></tr>
</table>

(2)　功能按钮

Open打开	加载选择的文件。加载后对话框关闭。
Cancel取消	不执行任何操作并关闭对话框
Help帮助	显示对话框在线帮助文件

环境设置文件保存对话框

这个对话框用来保存当前窗口的设置内容到一个设置文件(参考"5.15.3 窗口设置信息 (设置文件)".)

图 6-63：环境设置文件保存对话框

- 打开
- 每个区域的解释

打开

选择 [File] 菜单的[Environment]中的 [Save As...]选项保存当前窗口。

每个区域的解释

(1) 保存文件设置区域

Save in:保存至	这个区域用来指定文件名称。一个文件名可以直接输入，或者从这个区域上部的列表中选择。 扩展名最多可以指定257个字符的字符串。
File name:文件名	
Save as type:保存为某类型	这个区域用来指定保存文件的类型(扩展名) (参考"表5-22：设置文件类型".)　当前窗口中显示相应的文件扩展名称

(2) 功能按钮

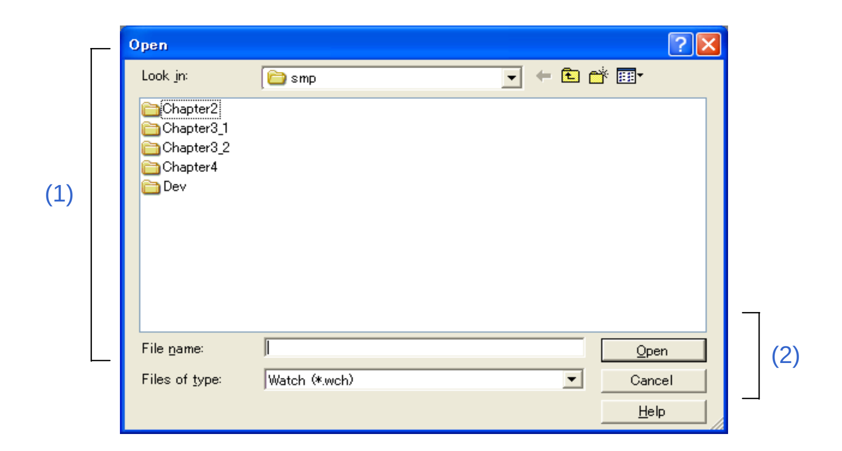
Save保存	保存当前窗口的设置信息到选择的文件。保存后关闭此对话框
Cancel取消	放弃修改并关闭对话框
Help帮助	显示对话框在线帮助文件

环境设置文件加载对话框

这个对话框用开读取设置文件 (参开"5.15.3 窗口设定信息 (设置文件)".)

当设置文件被加载后，目标窗口打开同时恢复保存的设置信息。

图 6-64：环境设置文件加载对话框



- 打开
- 每个区域解释

打开

选择[File] 菜单 [Environment] 中的[Open...]选项.

每个区域解释

(1)　加载文件设置区域

Look in:在某范围内查找	这个区域用来指定加载文件的名。可以通过键盘输入文件名，或从列表中选择。 扩展名最多可以指定257个字符的字符串
File name:文件名	
Files of type:文件类型	这个区域用来指定加载文件的类型(扩展名) (参考"表5-22 设置文件类型".)

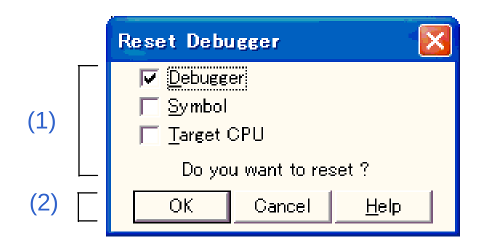
(2)　功能按钮

Open打开	加载选择的文件，加载之后关闭此对话框
Cancel取消	放弃修改并关闭对话框
Help帮助	显示对话框在线帮助文件

复位调试器对话框

这个对话框用来初始化ID850QB、CPU和符号信息。

图 6-65：复位调试器对话框



- 打开
- 每个区域的解释

打开

选择[File] 菜单中[Debugger Reset...]选项。

每个区域解释

(1)　选择区域的复位项目

这个区域指定需要初始化的选项。初始化选中的选项。

调试器	初始化ID850QB (缺省)。
符号	初始化符号信息
目标 CPU	初始化CPU.

(2)　功能按钮

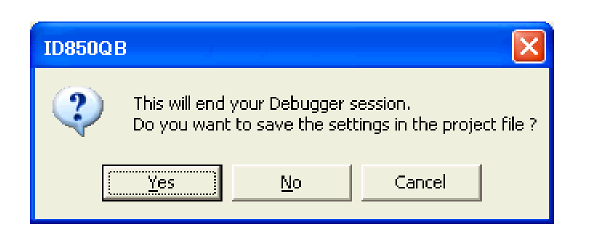
OK确认	根据设置初始化
Cancel取消	放弃修改，关闭此对话框
Help帮助	显示对话框在线帮助文件

退出调试器对话框

这个对话框用来选择退出ID850QB时当前调试环境是否被保存为一个工程文件(参考"5.15.1 调试环境 (工程文件)".)

可以在调试选项对话框中指定终止ID850QB而不打开此确认对话框。

图 6-66：退出调试器对话框



- 打开
- 功能按钮

打开

- 选择 [File] 菜单的[Exit]选项。
- 如果是强制终止，诸如应用程序终止，在终止窗口的任务列表上执行。

功能按钮

Yes是	保存当前的调试环境到一个工程文件，关闭所有窗口，终止ID850QB. 如果没有指定工程文件，将打开工程文件保存对话框。如果选择工程文件保存对话框中的<Cancel> 按钮， 调试环境既不被保存，ID850QB也不终止。 (在调试期间如果加载或保存了工程文件，缺省的焦点这个按钮上。)
No否	关闭所有窗口并终止ID850QB. (在调试期间如果没有加载和保存工程文件，这个按钮有缺省的光标)
Cancel取消	不执行任何操作关闭对话框

关于对话框

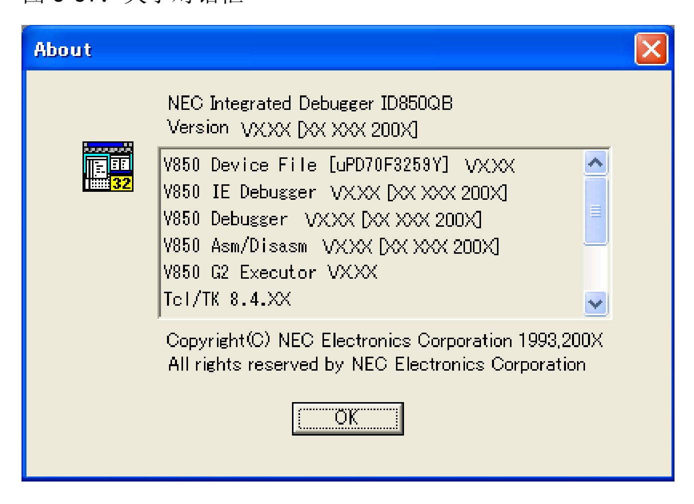
这个对话框显示ID850QB的版本信息(年份以4位数字显示)

备注: 通过选择[Select All and Copy (&C)]上下文菜单可以将对话框中的版本信息拷贝到剪贴板中。

下列的版本信息将被显示：

- ID850QB产品版本
- 设备文件版本
- GUI版本
- 调试器DLL版本
- 汇编器DLL版本
- 执行器版本
- Tcl/Tk版本
- 产品ID代码和在线仿真器的版本

图 6-67：关于对话框



- 打开

打开

选择 [Help] 菜单中 [About...]选项。

备注: 这个对话框也可在配置对话框中通过单击<About>按钮打开

控制台窗口

这个窗口用来输入ID850QB的控制命令.

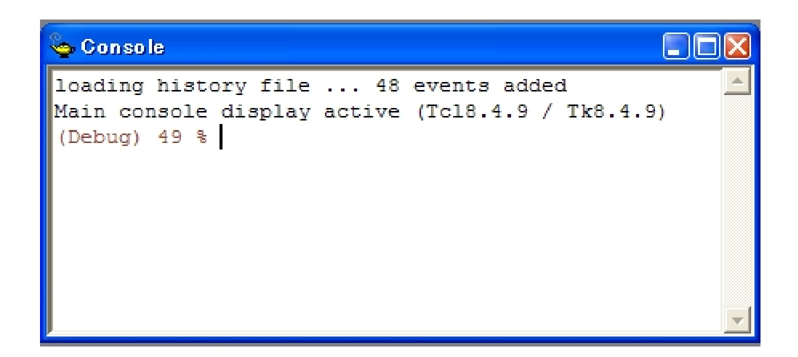
如果控制态窗口是激活状态，由于采用类Emacs键绑定，将不响应加速键。 但是F1 键可以在控制台窗口中打开在线帮助文件

当控制台窗口打开时，控制台窗口中显示一个只有<OK>按钮的错误信息窗口。

当控制台窗口关闭后，输入命令的历史记录将被保留。

关于更加详细的命令规范参考 "第7章命令参考" 。

图 6-68：控制台窗口



- 打开

打开

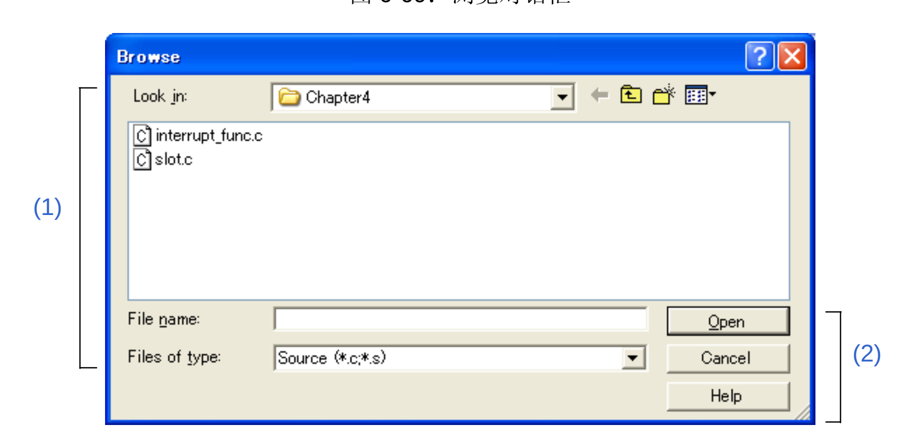
通过选择[Browse]菜单的 [Console]选项打开。

浏览对话框

这个对话框用来选择在源文本移动对话框中设定的文件

备注:　如果系统起动后首次打开此对话框，将显示第一次指定的源文件路径。当对话框是第二次或以后打开时，将保存和再次显示先前显示的目录。但是如果单击<Cancel> 按钮，将不保存先前显示的目录。

图 6-69：浏览对话框



- 打开
- 每个区域的解释

打开

在目标对话框中通过单击<Browse...>按钮打开。

每个区域的解释

(1)　打开文件设置区域

Look in:在某范围查找	这个区域用来指定打开文件的名称。可以从键盘直接输入文件名称，或从文件列表中选择。可指定的扩展名最多可达257字符。
File name:文件名	
Files of type:文件类型	这个区域用来指定打开文件的类型(扩展名)(参考"Table 5-5 可被显示的文件类型".)

(2) 功能按钮

Open打开	设置选中的文件。设置文件后关闭对话框
Cancel取消	不执行任何操作关闭对话框
Help帮助	显示对话框在线帮助文件

第 7 章　命令参考

这一章详细介绍ID850QB中的命令功能

- 命令行规则
- 命令列表
- 别名列表
- 变量列表
- 软件包列表
- 主键绑定
- 扩展窗口
- 回调过程
- 挂钩程序
- 相关文件
- 注意事项
- 命令解释
- 样本 (计算机脚本)

7.1　命令行规则

命令行规范详细规则如下:

- 为命令行指定名称，选项，参数
- 使用空格(Tab键或空格键)来区分单词
- 一行结束时使用一个换行符或者一个分号。
- 当一个指令名称和一个选项进入可识别点的时候,他们被识别出来。
- 在脚本中,必须输入指令名全称。

指令格式

```
命令　－选项　参数1　参数2　参数3 …
```

7.2　命令列表

表 7-1：调试器控制命令列表

指令名称	功能
address	地址表达式语句的赋值
assemble	反汇编/ 行汇编
batch	运行批处理(有回显)
breakpoint	设定/删除断点
Cache 缓存	设定高速缓存**[MINICUBE]**
dbgexit	中断 ID850QB
dbgopt	选择调试器选项
download	文件的下载
erase	清除内部闪存
extwin	创建扩展窗口
finish	从函数返回
flop	处理相关的内部闪存
go	连续运行
help	显示帮助
hook	挂起的设置
ie	显示/ 设定IE寄存器
inspect	变量检查
jump	跳转到窗口
map	设定 / 删除存储器图表
mdi	设置扩展窗口

指令名称	功能
memory	显示/设置存储器的设定
module	文件及函数的列表显示
next	程序步骤
refresh	重画窗口
register	显示/ 设定寄存器的数值和IOR 数值
reset	复位复位
run	复位复位并运行ＣＰＵ
step	单步运行
stop	停止运行
upload	上传
version	显示版本信息
watch	显示/设置变量
where	堆栈跟踪
wish	Tclet 的开始
xcoverage	覆盖率操作[IECUBE]
xtime	定时器操作 [IECUBE]
xtrace	跟踪操作 [IECUBE]

表 7-2 控制台 / Tcl 命令列表

指令名称	功能
alias	另外一个名称的创建
cd	改变目录
clear	清除荧屏
echo	响应
exit	关闭/结束
history	显示历史
ls	文件的显示
pwd	目录的检查
source	批处理执行
time	指令执行时间测量
tkcon	控制台控制
unalias	删除别名
which	显示指令路径或别名

指令名称	功能
Other	基于 Tcl/ Tk 8.4

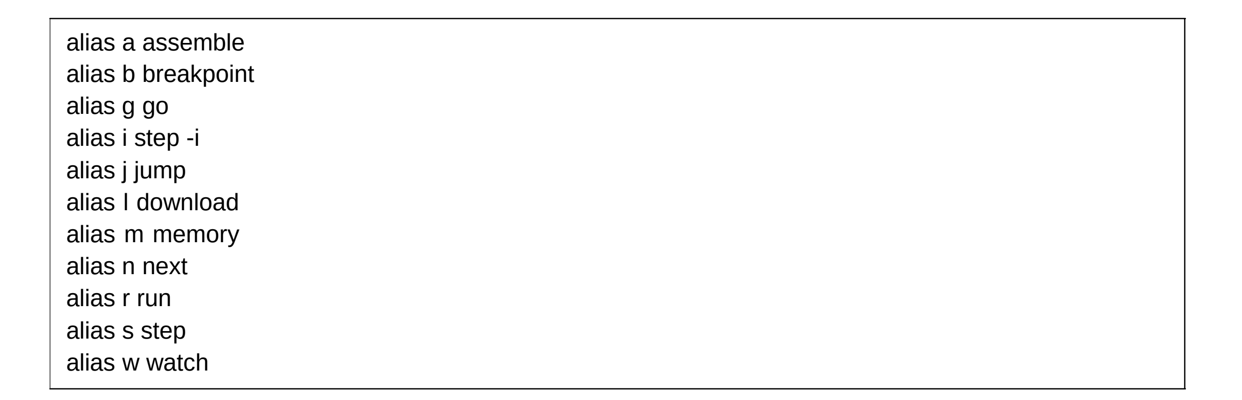
7.3　别名的列表

通过在" bin/idtcl/aliases.tcl"中定义命令的别名，可以用其它的名称来指定指令。

下面是默认的别名的描述。

可以使用编辑器编辑文件。

表 7-3　aliases.tcl 文件的内容



```
alias a assemble
alias b breakpoint
alias g go
alias i step -i
alias j jump
alias l download
alias m memory
alias n next
alias r run
alias s step
alias w watch
```

7.4　变量列表

表 7-4　变量列表

变量	功能
dcl (chip)	芯片名称只读
dcl (prjfile)	工程文件名只读
dcl (srcpath)	源路径只读
dcl (ieid)	IE类型只读
dcl (iestat)	IE 状态只读
dcl (bkstat)	中断状态只读
env (LANG)	语言
dcl_version	Dcl 版本只读

7.5　软件包的列表

表 7-5　软件包的列表

软件包	功能
tcltest	恢复测试
cwind	自动窗口控制
BWidget	工具箱
tcllib	Tcl库
mclistbox	多栏列表框
combobox	组合框

7.6　主键绑定

- 类似tcsh+ Emacs
- 指令名称的完成 [Tab]
- 文件名称的完成 [Tab]
- HTML 帮助 [F1]

7.7　扩展窗口

可以使用Tk建立扩展窗口。

在扩展窗口中，Widget被分派给'.dcl'作为一个后缀而不是'.'.

当下列的脚本文件被分配在bin/idtcl/tools/中时，一个扩展窗口被加到[Browse]菜单的->[Other]里面。

mdi 指令是扩展窗口一个独立的指令，被增加进来。

```
# Sample.tcl
wm protocol .dcl WM_DELETE_WINDOW { exit }
mdi geometry 100 50
button .dcl.b -text Push -command exit
pack .dcl.b
```

注意事项:　由于 MDI 窗口的限制原因，在扩展窗口中， Tk菜单指令不能够使用。

7.8　回调程序

扩展窗口能通过调用异步消息保持dcl_asyncproc程序。.

```
proc dcl_asyncproc {mid} {
    if {$mid == 19} {
        redraw
    }
}
```

异步消息ID 被传给dcl_asyncproc程序的参数

消息ID见下表:

表 7-6 消息 ID

消息ID	意义
9	变更配置之后
10	在登录事件之后
11	在删除事件之后
12	在运行之前
13	在断点之后
14	在复位复位CPU之后
15	在复位复位 ID850QB 之后
17	在更改扩展选择项之后
18	在更改调试器选择项之后
19	在下载之后
20	在更改存储器或寄存器之后
36	在开始跟踪之前[IECUBE]
37	在停止跟踪之后[IECUBE]
40	在开始定时器之前[IECUBE]
41	在停止定时器之后[IECUBE]
42	在清除跟踪之后[IECUBE]
45	在复位复位变量之后
46	在从 RRM功能 改变到跟踪功能之后，以及从跟踪功能改变到RRM功能之后[IECUBE]

7.9　挂钩程序

一个挂钩能使用挂钩程序在 ID850QB 中进行设定。

挂钩程序如下所示：:

- BeforeDownload(在下载之前的挂钩)
- AfterDownload(在下载之后的挂钩)
- AfterCpuReset(在断点过程中CPU复位复位之后的挂钩)
- BeforeCpuRun(在开始运行之前的挂钩)
- AfterCpuStop(在断点之后的挂钩)

通过使用挂钩程序，寄存器数值能在下载程序之前或在CPU复位复位之后被改变。

下面是一个实际的例子。 一个挂钩直到 ID850QB 被关闭前一直是有效的。

(1) [当挂钩与 ID850QB控制指令一起设定的时候]

1)　利用编辑器创建一个脚本文件a.。

2)　启动ID850QB，选择 [Browse浏览] 菜单 ->[Console控制台]，并且打开控制台窗口。

3)　如果脚本文件在下面窗口中已执行，那么在脚本文件中的挂钩已经被设定。

%hook test.tcl

(2) [当挂钩被在工程文件下载的时候被设置时]

1)　用编辑器产生脚本文件a。[注]

2)　启动 ID850QBand读取 test.prj 。 脚本文件的挂钩被设定。

```
proc BeforeDownload {}
    { register MM 0x7
    register PMC8 0xff
    register PMC9 0xff
    register PMCX 0xe0
}

proc AfterCpuReset {}
    { register MM 0x7
    register PMC8 0xff
    register PMC9 0xff
    register PMCX 0xe0
}
```

注:　确定脚本文件与工程文件同名。

例子:

与test.prj文件相对应的脚本文件名为test.tcl 。

将test.prj ， test.pri 和 test.tcl分配在相同的目录里面 。

7.10 关联文件

表 7-7 关联文件列表

文件名称	功能
aliases.tcl	当 aliases.tcl 控制台启动的时候运行。 设定默认的别名等等
Projectfilename.tcl	当工程文件 name.tcl [被打开的时候运行。 下列的挂钩可被使用。 BeforeDownload AfterDownload AfterCpuReset BeforeCpuRun AfterCpuStop
Loadmodulefilename.tcl	当加载模块 name.tcl 载入模块文件被下载的时候运行。 下列的挂钩能被用 BeforeDownload AfterDownload AfterCpuReset BeforeCpuRun AfterCpuStop

7.11 注意事项

- 文件和路径的分隔符是一条斜线。 (/)
- 当一个控制台是开启的时候,错误消息被输出到控制台。
- 为了强制地结束指令,关闭控制台。
- 外部指令 (DOS 指令) 的运行默认情况下是OFF状态。

7.12　指令的解释

在本节中，使用下面的格式解释每个指令。

指令名称

描述指令名称。

输入格式

描述指令的输入格式。

在下列的解释中，斜体字表示一个参数是可由用户给定，用"?"括起来的参数可以省略。

当一个指令名称和一个选项被输入到一个可识别的位置之后，它们是可被识别的。

别名

如果指令被定义了另外的一个名称，称为别名。详情请参考 "7.3 别名列表"..

功能

解释指令的功能。.

用法举例

显示指令用法一个例子。

地址

地址 – 地址表达式赋值

输入格式

address *expression*

功能

转换用表达式指定的地址表达式为地址。

用法举例

```
(IDCON) 1 % address main
0xaa
(IDCON) 2 % address main+1
0xab
```

汇编

汇编 - 反向汇编/ 行汇编

输入格式

assemble *?*options*? address ?code?*

功能

汇编由地址指定的代码所指定的特征字符串。

当 '.'为地址被指定,他被理解为前面汇编紧接的地址。

当代码被省略的时候,它从指定的地址字段中进行汇编。

下列各项是选择项：汇编时可以忽略：

-code	指令码也被显示。在汇编时可以省略。
-**number** *number*	*Number* 行被显示。在汇编时可以省略。

用法举例

```
(IDCON) 1 % assemble -n 5 main
0x000000aa   B7          PUSH   HL
0x000000ab   B1          PUSH   AX
0x000000ac   891C        MOVW   AX,SP
0x000000ae   D6          MOVW   HL,AX
0x000000af   A100        MOV    A,#0H
(IDCON) 2 % assemble main mov a,b
(IDCON) 3 % assemble . mov a,b
```

批处理

批处理 – 执行批处理，(带回显)

输入格式

batch scriptname

功能

在批处理中执行，同时通过指定的scriptname在荧屏上显示文件。

嵌套可以被使用。

用法举例

```
(IDCON) 1 % clear
(IDCON) 2 % batch bat_file.tcl
(IDCON) 3 % tkcon save a:/log.txt
```

断点

断点 - 设定/删除断点

输入格式

breakpoint *?options? ?address1? ?address2?*
breakpoint -delete *brkno*
breakpoint -enable *brkno*
breakpoint -disable *brkno*
breakpoint -information

别名

b

功能

通过指定的选项和地址操作断点。

如果一个断点能正确地被设定，返回断点号的值。

下面是选项:

-**software**	一个软件断点被指定。
-**hardware**	一个硬件断点被指定。 (默认值)
-**execute**	地址运行中断点被设定。 (默认值)
-**beforeexecute**	在地址运行前的断点被设定。
-**read**	一个地址数据读取断点被设置。
-**write**	一个地址数据写入断点被设置
-**access**	一个地址数据访问断点被设置
-**size** *size*	访问大小被设定。 (8,16 或 32)(单位: bit)[IECUBE]
-**data** *value*	数据条件被设定。
-**datamask** *value*	数据掩码被设定。
-**information**	断点的列表被显示。
-**delete**	指定断点号的断点被删除。
-**disable**	指定断点号的断点被禁用。
-**enable**	指定断点号的断点被使能。

用法举例

```
(IDCON) 1 % breakpoint main
1
(IDCON) 2 % breakpoint -i
1 Brk00001 enable rammon.c#17

(IDCON) 3 % breakpoint -software sub
2
(IDCON) 4 % breakpoint -i
1 Brk00001 enable rammon.c#17
2 Brk00001 enable rammon.c#8

(IDCON) 5 % breakpoint -disable 2
(IDCON) 6 % breakpoint -i
1 Brk00001 enable rammon.c#17
2 Brk00001 disable rammon.c#8

(IDCON) 7 % breakpoint -delete 1
2 Brk00001 disable rammon.c#8
```

高速缓存

[MINICUBE]

cache - 设定高速缓存

输入格式

cache

cache *config ?-icache itype? ?-dcache dtype?*

cache *clear ?-icache bool? ?-dcache bool?*

功能

当*config*被指定给子命令的时候,高速缓冲存储器类型被设定。

当*clear*被指定给子命令时，无论运行中是否清除高速缓存，都被设定(默认值是clear)

当子命令或下表被省略，显示当前状态。

itype 从下列各项中选择:

NB85E212	NB85E212 被使用.
NB85E213	NB85E213 被使用.
nouse	未使用 (默认).

*dtype*从下列各项中选择:

NB85E252	NB85E252被使用..
NB85E263	NB85E263被使用..
nouse	未使用 (默认).

Bool 从下列各项中选择:

0, false, or off	关闭
1, true, or on	开启

用法举例

```
(IDCON) 1 % cache config -i NB85E212 -d NB85E252
 (IDCON) 2 % cache
  i-cache: NB85E212
d-cache: NB85E252
(IDCON) 3 % cache clear -i false
(IDCON) 4 % cache
i-cache: NB85E212 (persist)
d-cache: nouse
```

dbgexit

dbgexit – 结束 ID850QB

输入格式

dbgexit *?options?*

功能

结束 ID850QB.

下列各项是选择项:

-**saveprj**	关闭 ID850QB时保存工程

用法举例

(IDCON) 1 % dbgexit -saveprj

dbgopt

dbgopt -选择调试器选项

输入格式

dbgopt *options ?value?*

功能

选择调试器的选项。 下面是选项::

-function *?func?* **[IECUBE]**	当 func 被省略的时候，切换 RRM 功能,trace功能 ，或coverage功能,当前的功能被显示。 *func* 从下列各项中选择:	
	rrm	RRM 功能被选择。
	Trace	Trace功能被选择。
	coverage	coverage功能被选择。

用法举例

(IDCON) 1 % dbgopt -function trace

下载

下载 - 文件的下载

输入格式

download *?options? filename ?offset?*

别名

l

功能

按照**选项**中指定的文件名下载文件(载入模块格式文件和 十六进制格式的文件自动辨认)。

如果**offset**被指定，地址将按偏移量进行移动 (如果数据格式为二进制,加载开始地址被指定为偏移量) 。

-binary	二进格式数据被下载。
-coverage	覆盖率数据被下载。
-append	附加的下载被运行。
-nosymbol	执行了下载。符号信息未读。
-symbolonly	符号信息被读。
-erase	内部的闪存内容在下载之前被全部擦除。(只对有内部闪存的产品.)
-reset	CPU在下载之后被复位复位。
-information	显示下载消息。

使用举例

(IDCON) 1 % download test.lmf

擦除

擦除 - 闪存的擦除

输入格式

erase

功能

抹除内在的闪存中的内容。

下面是选项:

如果没有一个下列的选择项被指定，擦除代码闪存中的内容。

-code	擦除代码闪存中的内容。
-data	擦除数据闪存中的内容。
-external	擦除外部闪存中的内容。

用法举例

(IDCON) 1 % erase -code -data

extwin

extwin- 扩展窗口的创建

输入格式

extwin scriptfile

功能

用 scriptfile 产生扩展窗口。

用法举例

(IDCON) 1 % extwin d:/foo.tcl

完成

finish - 从函数返回

输入格式

finish

功能

直到它返回到调用当前函数的程序的时候才执行。

使用举例

(IDCON) 1 % finish

flop

[MINICUBE]

flop –操作和用户有关的闪存

输入格式

flop -init

flop -user *clock*

功能

选择是否在用户闪存写入过程中，时钟相关设置是设备文件设置还是用户文件设置。

-init	使用设备文件设定执行过程。 如果设备文件没包含任何信息,将会产生一个内部闪存写入错误。 在这种情况下，使用 –user选择项。
-user *clock*	时钟产生函数的 IOR 被操作。 使用IOR为CPU时钟(fCPU)指定时钟频率(MHz) 在这种情况下，使用挂钩程序执行IOR设置。

使用举例

(IDCON) 1 % flop -user 13.5

go

go -连续运行

输入格式

go *?options?*

别名

g

功能

连续地运行程序。如果 -waitbreak 被指定，指令将等到程序停止运行。

下面是选项:

-**ignorebreak**	断点被忽略。
-**waitbreak**	指令等待程序停止运行。

使用举例

(IDCON) 1 % go -w

help

help -显示帮助

输入格式

help

功能

显示Dcl 帮助.

使用举例

```
(IDCON) 1 % help
For more information on a specific command, type HELP command-name
ASSOC       Displays or modifies file extension associations.
AT      Schedules commands and programs to run on a computer.
ATTRIB     Displays or changes file attributes.
BREAK    Sets or clears extended CTRL+C checking.
CACLS       Displays or modifies access control lists (ACLs) of files.
CALL     Calls one batch program from another.
CD      Displays the name of or changes the current directory.
CHCP     Displays or sets the active code page number.
CHDIR    Displays the name of or changes the current directory.
CHKDSK     Checks a disk and displays a status report.
CHKNTFS    Displays or modifies the checking of disk at boot time.
CLS      Clears the screen.
 :
 :
 :
 ::
 :
 :
```

hook

hook –挂钩的设定

输入格式

hook *scriptfile*

功能

使用scriptfile 为挂钩设定程序过程。

当工程文件被载入的时候，以及当ID850QB 复位复位的时候，挂钩设置被初始化。

使用举例

(IDCON) 1 % hook d:/foo.tcl

ie

IE 寄存器的显示/ 设定

输入格式

ie reg address *?value?*
ie dcu address *?value?*

功能

ie 指令依赖于 IE 。

当reg被指定给子命令的时候，IE寄存器的参考和设定被执行。

当 dcu 被指定给子命令的时候, DCU寄存器的参考和设定被执行。

注意:如果一个 DCU 寄存器被提到，寄存器的数值将会用0复位复位。

使用举例

(IDCON) 1 % ie reg 0x100 1
(IDCON) 2 % ie dcu 0x100 1

检查

inspect – 符号的检查

输入格式

inspect *?options? progname pattern*

功能

使用pattern的规则表达式指定 progname进行查找和显示载入模块的变量。

可以使用下列的规则表达式。

?	匹配 1个字符
*	匹配除了 0之外的字符
[chars]	匹配字符型字符。（ 指定范围 [a- z/0-9]都有可能.)
\x	匹配字符 x 。 (? *[]\ 也可以指定.)

下面是选项:

-nocase	不区分大小写。
-address	成对显示符号地址。

使用举例

(IDCON) 1 % inspect test1.out {[a-z]*}

跳转

jump – 跳转到窗口

输入格式

jump -source -line *filename ?line?*

jump *?options? address*

别名

j

功能

显示由 *options*. 指定的窗口

-**source**	源代码窗口从address指定地址处开始显示。
-**assemble**	汇编窗口从address指定地址处开始显示。
-**memory**	存储窗口从address指定地址处开始显示。
-**line**	将指令移到被*line*指定行。
-**focus**	焦点被移到显示的窗口。

使用举例

```
(IDCON) 1 % jump -s main
(IDCON) 2 % jump -s -l mainfile.c 10
(IDCON) 3 % jump -m array
```

映射

map - 设定/ 删除存储区映射

输入格式

map *options address1 address2 ?accsize? ?cs?*

功能

设定，删除 ，和显示存储区映射。

备注: 由 accsize 指定访问大小为8,16，或 32 .(单位:byte，默认值是 8)

注意事项: 为了用 ID850QB 映射一个存储区 (交替的ROM/RAM),可为芯片选择的 cs 指定cs0 ， cs1 ， cs2 ， cs3 ， cs4 ， cs5 ， cs6 或 cs7。
在 V850ES 系列中，因为芯片选择的分配可以被固化或者不提供芯片选择功能。这一指定可以被忽略。如果 cs 被指定，*accsize*的设定就不可以被省略。

下面是选项：

-erom [IECUBE]	交替的ROM被映射。 (由存储器板提供)
-eram [IECUBE]	交替的ROM被映射。 (由存储器板提供)
-target	目 标 区 域 被 映 射。
-targetrom [IECUBE]	目 标 ROM域 被 映 射。
-protect	I/O保护区域被映射。
-rrm	RRM 区域的起始地址被设定。 如果在用户程序执行阶段进行操作，CPU将会立即停止，RRM区域可以被分为8部分，起始地址和长度以列表的格式依下列各项被成对指定。 {{address size} {address size} {address size} ...} {{ 地址长度}{ 地址长度}{地址长度}.。。} 长度可以是256,512,768,1024,1280,1536,1792,2048个字节，而且总长度可以达到2048。
-clear	所有映射的设置被删除。
-information	指向映射的设定。

使用举例

```
(IDCON) 1 % map -i
1: 0 0x7fff 8 {IROM}
2: 0x8000 0x87ff 8 {Target RRM}
3: 0x8800 0x9fff 8 {Target}
4: 0xa000 0xf7ff 8 {NonMap}
5: 0xf800 0xfaff - {NonMap}
6: 0xfb00 0xfedf 8 {Saddr}
7: 0xfee0 0xfeff 8 {Register}
8: 0xff00 0xffff 8 {IOR}
(IDCON) 2 % map -erom 0x100000 0x10ffff
(IDCON) 3 % map -c
```

mdi

mdi - 扩展窗口的设定

输入格式

mdi geometry *?x y? width height*

mdi title *string*

功能

设定扩展窗口的大小和标题名称。.

指令只能被用于扩展窗口。

使用举例

```
(IDCON) 1 % mdi geometry 0 0 100 100
(IDCON) 2 % mdi title foo
```

存储器

存储器 - 显示/设置存储器

输入格式

memory *?options? address ?value?*
memory *?options?* -fill *address1 address2 value*
memory *?options?* -copy *address1 address2 address3*

别名

m

功能

在存储区中根据选项指定的地址设置地址值。

如果*value*被省略，显示存储区中由*address*指定的地址值。

如果 -fill被指定，从address1 到 address2的数据用*value* 进行填充。

如果 -copy被指定，将address1 到 address2 中的数据复制到address3 。

下面是选项:

-byte	以一个字节为单位进行显示/设置。 (默认值)
-halfword	以一个半字为单位进行显示/设置。
-word	以一个字为单位进行显示/设置。
-fill	填充数据。
-copy	复制数据。
-noverify	写入的时候，不执行校验。

备注: 如果在用户执行程序的过程中，下面任何一个操作被执行，CPU是瞬间停止，然后继续执行。

- 参考RRM 区域之外的一个存储器区域
- 内存的设置

使用举例

```
(IDCON) 1 % memory 100
0x10
(IDCON) 2 % memory 100 2
(IDCON) 3 % memory 100
0x02
(IDCON) 4 % memory -fill 0 1ff 0
```

模块

模块 - 文件及函数的列表显示

输入格式

module *progname ?filename?*

功能

显示由*progname*指定的加载模块文件与函数列表。

如果没有指定*finename*，显示文件列表。

如果指定了*finename*，则显示指定文件的函数列表。

使用举例

```
(IDCON) 1 % module rammon.lmf
1: rammon.c
(IDCON) 2 % module rammon.lmf rammon.c
1: rammon.c sub1
2: rammon.c main
```

Next

下一个 - 程序步骤

输入格式

next ?options?

别名

n

功能

运行程序步骤。如果函数被调用,执行完之后step停止。

下面是选项:

-**source**	指令在源行单元中运行。 (默认值)
-**instruction**	指令在指令单元中被运行。

用法举例

```
(IDCON) 1 % next -i
(IDCON) 2 % next -s
```

刷新

刷新-重画窗口

输入格式

refresh

功能

重画窗口并且更新数据。

使用举例

```
(IDCON) 1 % batch foo.tcl
(IDCON) 2 % refresh
```

寄存器

寄存器 – 显示/设置寄存器的值和 IOR 数值

输入格式

register *?*options*? regname ?value?*

功能

设置*regname*指定的寄存器的*value*。

如果*value*被省略,显示 *regname* 指定的寄存器的数值。

下面是选项:

-**force**	强制读或写正在运行。

备注: 如果在程序运行过程中，执行下面任何一个操作被，CPU瞬间停止，然后继续运行。

- 参考一个寄存器
- 设置一个寄存器

使用举例

```
(IDCON) 1 % register pc
0x100
(IDCON) 2 % register pc 200
(IDCON) 3 % register pc
0x200
```

复位复位

复位复位 - 复位复位

输入格式

reset *?*options*?*

功能

复位复位 ID850QB ，CPU，符号或事件。

如果选项被省略,CPU被复位复位。

下面是选项:

-cpu	CPU被复位复位。 (默认值)
-debugger	ID850QB 被复位复位。
-symbol	变量被复位复位。
-event	所有的事件和软件断点被复位。

使用举例

(IDCON) 1 % reset

运行

运行 - 复位并执行CPU

输入格式

run *?options?*

别名

r

功能

复位程序并且运行。

如果没有指定-waitbreak，指令直到程序终止也不等待。

下面是选项:

-waitbreak	指令等待程序停止。

使用举例

```
(IDCON) 1 % run
(IDCON) 2 % run -w
```

单步

单步 – 单步运行

输入格式

step *?*options*?*

别名

s(单步 -源)

i(单步 -指令)

功能

执行单步运行。

如果函数被调用，指令在函数的头部停止。

下面是选项。

-source	指令在源代码行单元中被运行。 (默认值)
-instruction	指令在指令单元中被运行。

使用举例

```
(IDCON) 1 % step -i
(IDCON) 2 % step -s
```

停止

停止 - 停止运行

输入格式

stop

功能

强制停止程序。

使用举例

```
(IDCON) 1 % run
(IDCON) 2 % stop
```

tkcon

tkcon -控制台控制

输入格式

tkcon cmd ?arg?

功能

控制控制台窗口。

这个指令是Console/Tcl 指令中的一个。(参考 "表7-2 Console/Tcl 指令列表")。

tkcon buffer ?size?	设定和参考控制台最大的缓冲区大小(行的数目)。 如果超过指定的缓冲区大小,多余的行将从原来的次序中删除。
tkcon close **tkcon destroy**	关闭控制台窗口。
tkcon font ?fontname?	设定和参考用于控制台窗口的字体。
tkcon gets	执行标准的输入比如： Stdin。 打开一个对话框。
tkcon history ?-newline?	显示指令历史。
tkcon save ?filename? ?type	把控制台窗口中的缓冲区数据存为一个文件。当文件名称或文件类型被省略的时候,打开一个对话框。 从所有的类型,history， stdin ，stdout, 和 stderr中选择。
tkcon version	显示控制台版本。

使用举例

(IDCON) 1 % tkcon save c:/temp/logfile.txt all

上传

上传 - 上传

输入格式

upload *?options? filename address1 address2*
upload -coverage *filename*

功能

将一个指定范围中的存储数据保存到文件中。

下面是选项:

-binary	数据被以二进制的格式保存。
-coverage [IECUBE]	覆盖率数据被保存。 当保存覆盖率数据的时候，所有指定范围数据被存入一个文件。 (不要求指定开始/ 结束地址的)
-intel	数据被以intel十六进制的格式保存。 (默认值).
-motorola	数据被以摩托罗拉的十六进制的格式保存。
-tektronix	数据被以Tektronix的十六进制的格式保存。。
-force	文件被覆盖。

使用举例

(IDCON) 1 % upload -b foo.hex 0 0xffff

版本

版本 - 显示版本消息

输入格式

version

功能

显示 ID850QB 的版本。

使用举例

(IDCON) 1 % version
GUI : V*x.xx* [*XX XXXX* 200*X*]
Devicefile : V850 Device File [uPD703201] V*x.xx*
Debugger : V850 Debugger V*x.xx* [*XX XXXX* 200*X*]
Executer : V850 Executer V*x.xx* [*XXX XXXX* 200*X*]
Monitor : V850 Peripheral V*x.xx* [*XX XXXX* 200*X*]
Assembler : V850 Asm/Disasm V*x.xx* [*XX XXXX* 200*X*]
Tcl/Tk : 8.4.12

观察

观察-显示/设置变量

输入格式

watch *?options? variable ?value?*

别名

w

功能

显示和设置变量。

下面是选项:

-binary	以二进制数显示数值。
-octal	以八进制数显示数值。
-decimal	以十进制数显示数值。
-hexdecimal	以十六进制数显示数值。
-string	以字符型显示数值。
-sizeof	以十进制数显示变量的大小，而不是变量值。
-encoding *name*	指定在字符串显示期间进行编码。 默认使用系统编码。name(编码名称)根据Tcl说明来进行。(shiftjis, euc-jp等)

使用举例

```
(IDCON) 1 % watch var
0x10
(IDCON) 2 % watch -d var
16
(IDCON) 3 % watch array\[0\] 0xa
```

Where

Where - 堆栈跟踪

输入格式

where

功能

运行堆栈的back-trace。

使用举例

```
(IDCON) 1 % where
1: test2.c#sub2(int i)#13
2: test.c#num(int i)#71
3: test.c#main()#82
```

Wish

wish - 启动Tclet

输入格式

wish scriptname

功能

使用 Tk(Tclet) 启动脚本。

可以使用Tclet创建扩展窗口。

使用举例

(IDCON) 1 % wish test.tcl

xcoverage

[IECUBE]

xcoverage –覆盖率操作

输入格式

xcoverage *option*

功能

操作覆盖率。

下面是选项:

-clear	清除覆盖率存储器。

使用举例

(IDCON) 1 % xcoverage -clear

xtime

[IECUBE]

xtime -定时器的操作

输入格式

xtime *option*

功能

操作定时器。

下面是选项:

-start	运行程序时启动定时器。
-stop	运行程序时停止定时器。
-gobreak	从运行 到 断点的时间以纳秒的形式显示。

使用举例

```
(IDCON) 1 % xtime -start
(IDCON) 2 % xtime -stop
```

xtrace

[IECUBE]

xtrace –跟踪操作

输入格式

xtrace -dump *?-append? frameno ?filename?*
xtrace -start
xtrace -stop
xtrace -clear
xtrace -*addup ?bool*?
xtrace -mode *?mode?*
xtrace -*complement ?bool*?

功能

跟踪操作。.

下面是选项:

-start	Tracer在运行程序的时候启动。
-stop	Tracer在运行程序的时候停止。
-clear	清除跟踪存储器。
-dump	跟踪的数据被处理(默认值) 。 dump结果被重新传入到控制台窗口。 如果文件名称被指定,dump结果将被存入该文件。
-append	将dump结果附加到一个文件。
-addup *?bool?*	选择时间标签是否是全部或者没有。 当 *bool* 被省略的时候，显示当前模式。
-mode *?mode?*	Trace控制模式 （ 任何一个:all，cond ， nonstop，fullstop，fullbreak ， delaystop ， delaybreak ，machine 或event)任选。 当mode被省略的时候，显示当前模式。
-complement *?bool?*	选择是否进行跟踪补偿。 当 bool 被省略的时候,显示当前模式。

使用举例

```
(IDCON) 1 % xtrace -start
(IDCON) 2 % xtrace -stop
(IDCON) 3 % xtrace -dump 3
_ 01685 2 000000BC M1 br _sub2+0x2
_ 01686 4 0000009A BRM1 st.w r6, 0x8[sp]
_ 01687 3 0000009E BRM1 st.w r0, 0x0[sp]
(IDCON) 4 % xtrace -clear
(IDCON) 5 % xtrace -addup true
```

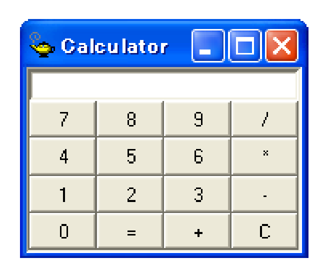
7.13　样例 (计算机脚本)

计算机脚本描述的扩展窗口的脚本和它的运行脚本屏幕显示如下：

扩展窗口的执行脚本

```
# Calculator.tcl
mdi geometry 100 100
set top .dcl
entry $top.e -relief sunken -textvariable v
frame $top.f -height 120 -width 120; pack $top.e -fill x; pack $top.f -fill both -expand 1
set i 0; set v {}; set r 0.25
foreach n {7 8 9 / 4 5 6 * 1 2 3 - 0 = + C} {
	if {$n == "=" || $n == "C"} {
		button $top.f.b$n -text $n
	} else {
		button $top.f.b$n -text $n -command "$top.e insert end $n"
	}
	place $top.f.b$n -relx [expr ($i%4)*$r] -rely [expr ($i/4)*$r] -relw $r -relh $r
	incr i
}
bind $top.f.bC <1> {$top.e delete 0 end}
bind $top.f.b= <1> {catch {expr $v} v}
```

图 7-1 运行屏幕显示



附录 A 扩展窗口

A.1 概述

通过ID850QB ，用户能在现有的窗口之外创建定制的窗口。

Tcl(工具命令语言) 译释机和控制调试器的指令在 ID850QB 中被实现。用户可使用 Tcl 创建窗口。

ID850QB 提供下列的扩展窗口的样本。

A.2 样本窗口

表 A-1 扩展窗口列表 (样本)

窗口名称	功能
List window	显示源文件及函数列表。
Grep window	搜索一个字符串。
RRM window	实时RAM监视的存储窗口。
Hook window	设定挂钩程序。
Memory Mapped I/O window	从指定的地址读入或写入数据。
Sym Inspect window	在一列适当描述变量中搜索。
Run Break Time window	显示两类的时间:用户程序开始运行的时间和产生断点的时间 。

A.3 激活

扩展窗口可以通过从[Browse浏览]菜单中的[Others其它]菜单中选择List, Grep, RRM, Hook, Memory, SymInspect or RunBreak- Time来激活。

备注: 每个.tcl 文件被安装在 NEC electronics tool\ ID850QB\Vx.xx\bin\idtcl\tool

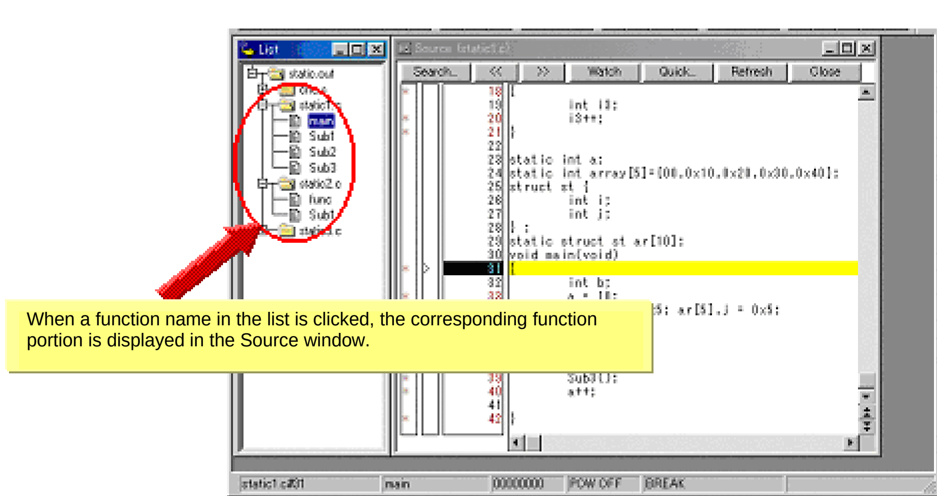
A.4 每个样本窗口的解释

ID850QB 提供下面的样本窗口。

列表窗口

源文件和函数列表以树型格式显示在这个窗口中。当一个窗口中的功能函数被点击后，显示相应的源文件。

图 A -1 列表窗口

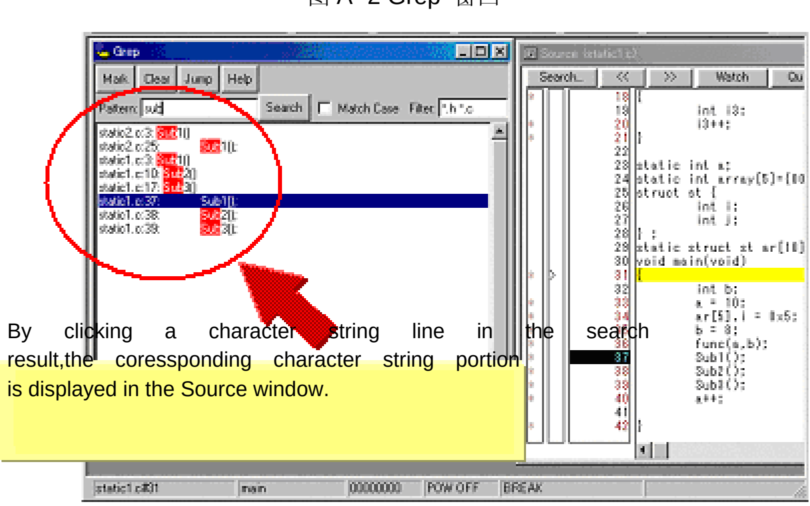


Grep 窗口

在源路径下运行的文件中查找一个字符串。

当搜索结果被点中之后,相对应的来源被显示出来。

图 A -2 Grep 窗口



对象	功能
模式	输入要搜索的字符串。
<Mark>按钮	标记被查找的字符串。
<Clear>按钮	清除标记。
<Jump>按钮	把光标放在查找结果中的一项上，并且点击此按钮以打开对应的文件。
Match Case	选择是否区别大写字母的和小写字母。
Filter过滤器	指定被查找的文件类型。

RRM 窗口

这是一个专门监视RAM的窗口。

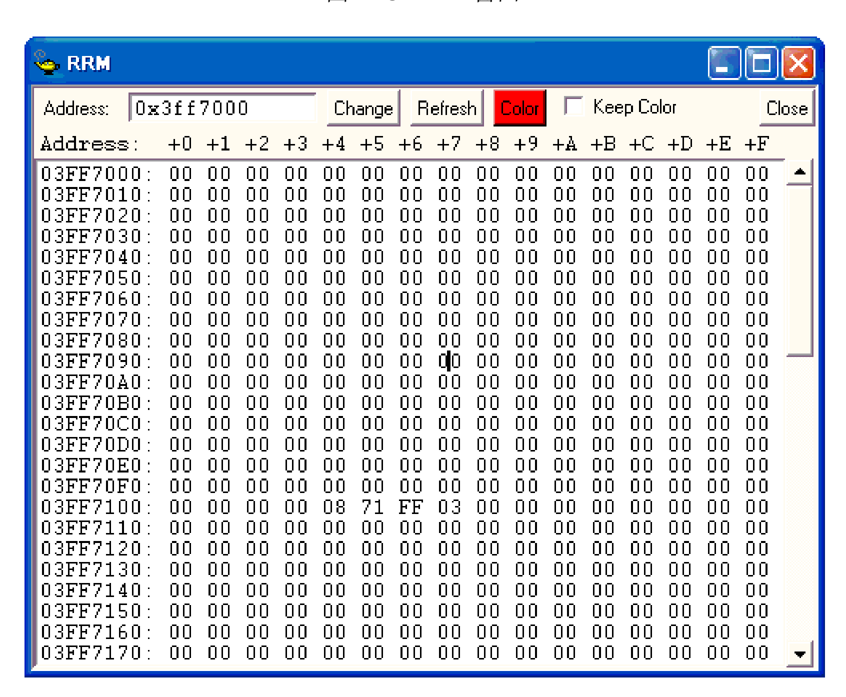
在程序执行过程中，RAM区域中值发生改变的地址区域以一种颜色高亮显示。 显示范围是 1 KB 。由于要读取RAM，用户程序的运行瞬间停止。在这个窗口上，当用户程序正在执行时，RAM区域的起始地址可被改变。

注意事项: 所有的数据没被同时读取(因为 1 KB 的数据是分开的，且以字为单位进行读取)

备注1: 即使在扩展选项对话框中RAM监视器功能被设定成OFF状态时，RRM 窗口仍被打开。

备注2: 采样间隔大约 0.3- 0.7 秒 (20 MHz), 但是它依赖于CPU运算频率。 **[MINICUBE]**

图 A -3 RRM 窗口



对象	功能
地址	输入要显示的开始地址 (自动对齐到 1 KB)
<Change>按钮	切换开始地址显示。
<Refresh>按钮	从存储器中读取数据。
<Color> 按钮	颜色可根据用户需要定制。 默认的颜色是红色。
保存彩色	指定是否保持颜色高亮。选中：一旦值改变了，颜色的高亮显示被保留到一个断点的出现。 未选中：如果没有数值的变化，颜色会被清除。 (默认值)
<Close>按钮	关闭这个窗口。

挂钩窗口

这个窗口是通过使用一个挂钩程序来为调试器设定挂钩的。

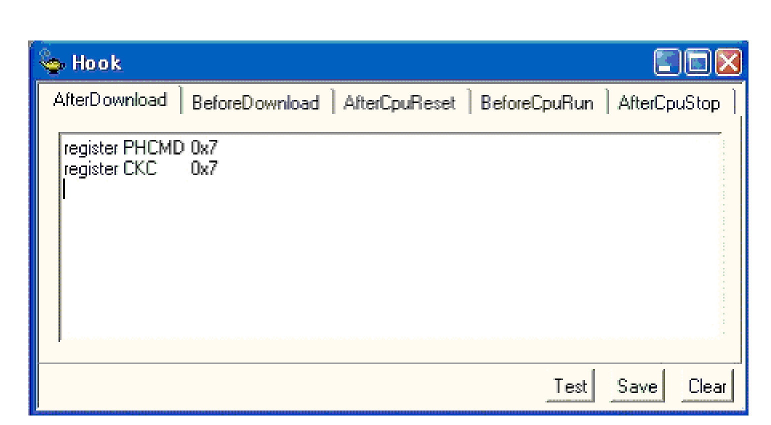
在下载一个程序之前或者CPU复位之后，挂钩程序使能改变寄存器数值。 在这个窗口上，一个挂钩能被通过使用下列的五个标签页进行设定：

- [AfterDownload]　标签页：在下载之后的挂钩
- [BeforeDownload] 标签页：在下载之前的挂钩
- [AfterCpuReset] 标签页：断点期间在CPU复位之后的挂钩
- [BeforeCpuRun] 标签页：开始运行前的挂钩
- [AfterCpuStop]　标签页：在断点之后的挂钩

备注:　在下载加载模块前，通过使用[BeforeDownload] 标签页来设置一个IOR，例如：可以高速下载。利用这个标签页也可以用来访问外围存储器。

如果设置以"project-file-name.tcl"被保存在工程文件存储的目录中的，下次打开工程的时候，将执行该设置。

图 A -4 挂钩窗口



对象	功能
[AfterDownload] 标签页	下载之后的挂钩 在下载被执行之后，输入到标签页的寄存器数值被指定的数值自动地覆盖。
[BeforeDownload]标签页	下载之前的挂钩 在下载被执行之前，输入到标签页的寄存器数值被指定的数值自动地覆 盖。
[AfterCpuReset] 标签页	断点期间在CPU复位之后的挂钩 在复位CPU之后，输入到标签页的寄存器数值被指定的数值自动地覆盖。
[BeforeCpuRun] 标签页	运行启动前的挂钩 在开始运行之前，输入到标签页的寄存器数值被指定的数值自动地覆盖。。
[AfterCpuStop] 标签页	中断之后的挂钩 在中断之后，输入到标签页寄存器数值被指定的数值自动地覆盖。。
<Test>按钮	在标签页上的所有指令被测试。
<Save>按钮	把所有标签页内容保存到一个文件中。 如果 ID850QB 从一个工程文件中被激活，该文件被保存为"project-filename.tcl"
<Clear>按钮	清除在标签页上的所有描述符。

备注: 用程序寄存器和外围I/O寄存器指定寄存器名。

内存映射 I/O 窗口

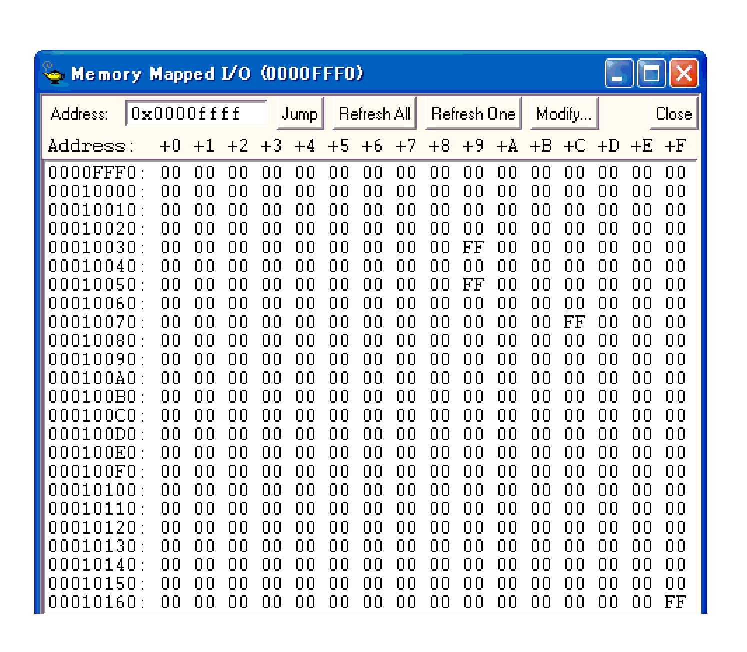
数据明确地在这个窗口的一个指定的地址中被读写。

当一个写操作在存储窗口中被执行的时候，数据 在ID850QB内部读取并且校验。 除此之外，存储器也可以简单的通过滚动存储窗口进行读取。然而在存储区映射I/O窗口上，上述的操作没有执行。

因此，这个窗口对读或写一个特定的地址是很有用的。

当用户程序正在运行时，在数据被写入这个窗口之前瞬间停止。。

图 A -5 内存映射 I/O 窗口



对象	功能
地址:	输入要显示的地址。 显示对象地址修改可由按Enter键或通过点击 <Jump>按钮。数据内容在此时没被读取，因此地址 (数值) 被显示在地址中显示区段，但是 " XX" 在数据区段中被显示。
<Jump>按钮	跳转到地址栏中输入的地址。
<Refresh All>按钮	只读取目前所有显示区域一次。 当企图读取一个未被映射的区域，或当读取时发生了一个错误的时候， " ZZ" 将会在数据区段中被显示。
<Refresh One>按钮	读取存储区中光标仅放置过一次的地址处的数据读取数据量的多少依赖显示格式。当企图读取一个未被映射的区域，或当读取时发生了一个错误的时候， " ZZ" 将会在数据区段中被显示。
<Modify…>按钮	打开内存映射I/O对话框。 光标放置的地址是显示在存储映射I/O对话框中的输入地址。如果这个按钮在存储映射I/O对话框中光标位置改变后被按下，存储映射I/O对话框中的地址字段也将改变。

对象	功能
<Close>按钮	关这个窗口。
上下文菜单	从Byte， HalfWord 和 Word类型中选择显示格式。

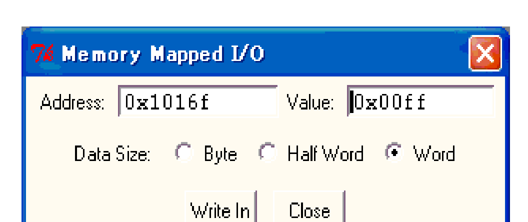
存储映射 I/O 对话框

这个对话框可以通过点击一下存储映射I/O窗口上的<Modify… >按钮打开。它用于对任何地址写入数据。

备注1: 当被写入数据的区域显示在存储窗口或者观察窗口中的时候，在<Write in>按钮被点击后，该数据将被这些窗口读取。

备注2: 如果数据长度小于配置对话框中指定的访问长度，ID850QB 以访问长度对数据进行一次读取，改变读取数据的对应部分，并且以访问长度写入被改变的数据。

图 A -6 内存映射 I/O 对话框



对象	功能
地址:	输入要写入的地址。 默认显示存储映射I/O窗口光标所在的数据相应的地址
数值:	输入要写入的数值。
数据长度:	选择被写入的数据长度。 以默认的方式选择存储映射I/O窗口中指定的长度。
<Write In>按钮	以指定长度在指定地址写入数据。
<Close>按钮	关闭这个对话框。

Sym 检查窗口

这个窗口显示符号列表和加载模块文件的地址，而且被用于从列表查找合适的描述符。

图 A -7 Sym 检查窗口

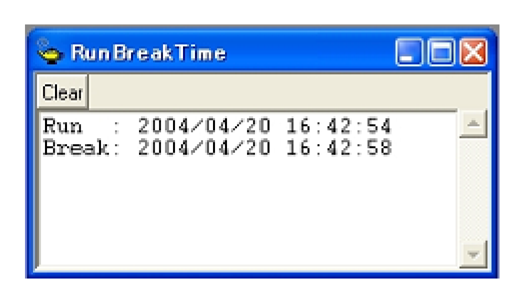
对象	功能
加载模块:	选择一个加载模块文件。
过滤器:	指定一个适当的描述符，以便找回该变量。
匹配 大小写	在过滤器中，指定是否区分大小写， 检查这个对话框来区分大小写的敏感性。
符号	显示符号。 点击这个以字母顺序排序存储的变量图标。
地址	显示地址。点击这个以数字升序排序存储的地址图标。

上下文菜单	功能
Copy	将选择的地址复制到剪切板上。
Jump to Source	从选中行上的地址跳转到源文件窗口中显示的同一地址上。
Jump to Assemble	从选中行上的地址跳转到汇编窗口中显示的同一地址上。
Jump to Memory	从选中行上的地址跳转到存储窗口中显示的同一地址上。

运行断点时间窗口

这个窗口显示两类时间:用户程序开始运行的时间和产生断点的时间。 当测量时间很长的时候，这个窗口是很有帮助的。 Windows 定时器功能被用到这个窗口上，时间以小时、分钟和秒的形式显示。

图 A -8 运行断点时间窗口



对象	功能
<Clear>按钮	清除时间显示

附录 B 输入规则

B.1　可用字符集

表 B-1 字符集列表

分类	字符
字母字符	大写： A B C D E F G H J K L M N O P Q R S T U V W X Y Z 小写: a b c d e f g h i j k l m n o p q r s t u v w x y z
数字	0123456789
与字母等同的字符	@ _

表 B-2 特殊字符列表

字符	名称	用法
(	左括弧	改变运算符顺序。
)	右括弧	改变运算符顺序。
+	加号	加法运算符或正号
-	减号	减法运算符或负号
*	星号	乖法运算符或间接引用运算符
/	斜线	除法运算符
%	百分比	取余运算符
~	取补运算符	取补运算符
|	竖线	位或运算符
^	异或	异或运算符
&	地址符	位与运算符或者地址运算符
	句号	直接成员运算符或者小数点
,	逗号	分隔运算符
[	左中括号	数组运算符或者基址寄存器指定符号
]	右中括号	

B.2　变量

(1)　一个变量由下列任何字符组成。

A到 Z, a到 z,@,_(下划线),?, 以及 0 到 9

(2)　一个变量必须以除了数字0 到 9.之外的字符开头

(3)　大小写字母是有区别的。

(4)　一个符号必须不超过 2048个字符长。 (如果一个超过 2048个字符的符号被定义, 只有前面 2048个字符是有效的)

(5)　一个变量由加载的加载模块文件定义。

(6)　变量通过有效范围可分为下列类型：

全局变量 (汇编语言, C语言)
静态变量 (C语言)
- 在文件中的静态变量
 - 在函数中的静态变量
局部变量 (C语言)
- 在文件中的局部变量
- 在函数中的局部变量
-在区块中的中的局部变量

(7)　下列的符号可用于每种语言:

汇编语言，结构化汇编语言
标识名称,位元符号名称
C语言
变量名 (包括指针变量名，枚举类型变量名，数组变量名，结构体变量名，以及共用体变量名
函数名称，标识名称
数组元素，结构体元素，共用体元，位元 (如果符号是一个数组，结构体或者共用体)

(8)　一个符号能替换一个地址或数字的数值。

(9)　当源文件经过汇编或者编译之后，符号的有效范围在源调试信息的基础上决定。

(10)　只描述全局变量名称。

(11)　一个局部变量和一个文件名成对出现。

B.3　数值

可以使用下列的四类型的数值。 每个类型的输入格式如下所示。

后缀 (粗体) 和十六进制的字母字符可以是大写字母，也可以是小写字母。 如果第一个字符是A到F,0必须是它的前缀。

在 ID850QB 的输入栏，根据默认数制，十进制或十六进制数交替地被选择。

表 B-3 数值输入格式

数字的数值	输入格式
二进制数	nY n...nY (n=0,1)
八进制数	nO n...nO (n=0,1,2,3,4,5,6,7) nQ n...nQ (n=0,1,2,3,4,5,6,7)
十进制数	n n...n nT n...nT (n=0,1,2,3,4,5,6,7,8,9)
十六进制数	n n...n nH n...nH 0xn 0xn...n (n=0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)

B.4　表达式和运算符

(1)　表达式

一个表达式包括常量，寄存器名称，周边I/O寄存器名称和通过运算符连接起来的符号。

如果周边I/O寄存器名称,标识名称，函数名称,或变量名称被描述为一个符号，并且一个地址被计算出来作为符号的值。

除了运算符外，组成一个表达式语句的元素被称作项(常量和标识) ，从左到右分别被叫做第一项式，第二项，依此类推。

(2)　操作符

下面的操作符可以被C语言使用:

表 B-4 操作符列表

变量	意义	解释
算术运算符		
+	加	返回第一个和第二个项之和。
-	减	返回第一个和第二个项之差。
*	乘	返回第一个和第二个项之积。
/	除	返回第一个和第二个项之商，，并返回结果的整数部分。
MOD %	取余	返回第一个和第二个项相除之后的余数。
- sign	一元操作符 (负号)	返回项的二进制补码值。
+ sign	一元操作符 (正号)	返回项的数值。
逻辑运算符		
NOT ~	取反	取每个逻辑位的反并返回结果。
AND &	与	取两个项中每个位逻辑积，并返回结果。
OR |	或	取两个项中每个位逻辑和，并返回结果。
XOR ^	异或	取两个项中每个位异或逻辑和，并返回结果。
移位运算符		
SHR >>	右移	将第一个项向右移第二个项中的值所代表的位数，并返回结果。并以相同个数的0替换移过后的高位。

Symbol	意义	解释
SHL <<	左移	将第一个项向左移动第二个项中的值所代表的位数，并返回结果。并以相同个数的0替换移过后的低位。
字节分割符		
HIGH	高位字节	返回该项最低16位中的高8位。
LOW	低位字节	返回该项最低16位中的低8位。
字分割符		
HIGHW	高字节	返回该项最低32位中的高16位。
LOWW	低字节	返回该项最低32位中的低16位。
其他		
(	左括弧	先执行括号里面的运算符，再执行其外面的运算符。 '(' 和 ')' 总是成对出现的
)	右括弧	

(3)　运算符规则

运算符依照其优先级进行先后执行。

表 B-5 运算符的优先级

优先级	运算符
1　　高	(,)
2	+ sign, - sign, NOT, ~, HIGHT, LOW, HIGHW, LOWW
3	*, /, MOD, %, SHR, >>, SHL, <<
4	+, -
5	AND, &
6　　低	OR, |, XOR, ^

- 如果运算符的优先级次序是一样的,运算从左向右执行。
- 先执行( )里面的，再执行( )外面的。
- 每一个项被当做无符号32位数据。
- 所有运算结果被当做无符号32位数据。
- 如果运算期间溢出，低32位有效，而且溢出没被检查出来。

(4)　项

为了描述项的一个常量,可以用下列的数值进行描述。

表 B-6 数制范围

数制	范围
二进制数	0Y <= 值 <= 11111111111111111111111111111111Y (32 位)
八进制数	0O <= 值 <= 37777777777O
十进制数	-2147483648 <= 值 <= 4294967295 (一个负十进制数被内部转换成二进制数的补码。)
十六进制数	0H <= 值 <= 0FFFFFFFFH

B.5　文件名称

下列规则适用于源文件名称和加载模块文件名称。

(1)　源文件名称和加载模块文件名称

文件名称由a到z, A到 Z,0 到 9,.,_,+, 和 -. 组成。

文件名称必须以除"."外的字符开头。

文件名称不能够用(.)或者空格作为其前缀或后缀。

文件名称与大小写无关。

包括路径，文件名称最多只能由259个字符组成。

(2)　其他的文件名称

其他的文件名称遵从 Windows 文件名称命名规则。

下列的字符不能够用于文件名中：

\/: *? "<>|;

文件名称不能够用(.)或者空格作为其前缀或后缀。

文件名称与大小写无关。

包括路径，文件名称最多只能由259个字符组成。

附录 C　键功能列表

表 C-1 键功能列表

键名	功能
BackSpace	删除光标前一个字符并且移动光标到删除的字符位置。 操作之后，在光标之后字符串整体向前移动。
Delete	-删除光标后一个字符并且字符串整体向前移动。 - 在事件管理器或者每个事件对话框中选择删除一个多样事件条件。 - 在视窗中删除选中的数据。
Insert	在源代码窗口和汇编窗口中交替地切换为插入或者改写模式。 然而，这个键在存储区、寄存器和 IOR 窗口中无效，并且只有改写模式可被用作输入模式。
PrintScreen	将整个屏幕以位图的格式装入剪切板(Windows的功能)
Esc	- 关闭下拉菜单。 -关闭模式对话框。 -恢复输入数据。
Alt	将光标移动到菜单条。
End	将光标移动到行尾。
Home	将光标移动到一行开头位置。
PageUp	将屏幕向上滚动一屏。 光标也移动到屏幕的顶部。
PageDown	将屏幕向下滚动一屏。 光标也移动到屏幕的顶部。
Space	插入一个空格符。
Tab	移动光标到下一制表栏。
Up arrow key	向上移动光标 如果光标在荧屏的底部，屏幕同时向上滚动一行。
Down arrow key	向下移动光标 如果光标在荧屏的顶部，屏幕同时向下滚动一行。
Right arrow key	向右移动光标 如果光标在荧屏的最左边，屏幕同时向右滚动一列。
Left arrow key	向左移动光标 如果光标在荧屏的最右边，屏幕同时向左滚动一列。
Enter	- 设定输入数据。 - 按默认的按钮。
F1	打开帮助窗口。

键	功能
F2	强制地停止程序运行。 与[Run]菜单中的[Stop]功能相同.
F3	复位CPU。 与[Run]菜单中的[CPU Reset]功能相同.
F4	复位CPU并运行程序。 与[Run]菜单中的[Restart]功能相同.
F5	运行程序。 与[Run]菜单中的[Go]功能相同.
F6	运行来源代码窗口或汇编窗口中光标位置的程序。与[Run]菜单中的[Come Here]功能相同.
F7	用户程序是实时运行直到运行返回。与[Run]菜单中的[Return Out]功能相同.
F8	单步运行程序。与[Run]菜单中的[Step In]功能相同.
F9	在光标位置设置源代码窗口或汇编窗口中源代码的断点。与[Run]菜单中的[Break Point]功能相同.
F10	下一步运行。与[Run]菜单中的[Next Over]功能相同.
F11	设定或删除一个软件断点。 与[Run]菜单中的[Soft Break Point]功能相同.
Shift+End	将选择范围扩展到行末。
Shift+Home	将选择范围扩展到行头。
Shift+Left arrow key	将选择范围扩展到左边一个字符。
Shift+Right arrow key	将选择范围扩展到右边一个字符。
Shift+F6	从当前光标位置运行源代码窗口或汇编窗口中的程序。与[Run]菜单中的[Start From Here]功能相同.
Shift+F9	复位CPU。 与[Run]菜单中的[CPU Reset]功能相同.
Ctrl+End	显示最后一行。 光标也将会移到最后一行。
Ctrl+Home	显示第一行。 光标也将会移到第一行。
Ctrl+Left arrow key	向左移动光标一个字。 如果光标在荧屏的最左边，屏幕同时向右滚动一列。。
Ctrl+Right arrow key	向右移动光标一个字。 如果光标在荧屏的最右边，屏幕同时向左滚动一列。。
Ctrl+F5	忽略断点的设定并运行程序。与[Run]菜单中的[Ignore break points and Go]功能相同.
Ctrl+F9	设定源代码窗口或汇编窗口的当前光标位置地址到PC。 与[Run]菜单中的[Change PC]功能相同.
Ctrl+A	选择事件管理器中所有登录的事件。 与事件管理器中的[View]菜单中的[Select All Event]功能相同。
Ctrl+C	复制选定的字符串，并存于剪切板缓冲区中。

Key	功能
Ctrl+D	反汇编，并显示跳转目标地址的结果，该地址由当前窗口中的选定的数据值来指定。打开汇编窗口。与 [Jump]菜单中的[Assemble]功能相同。
Ctrl+E	当PM+在运行时，通过PM+指定的编辑器中打开显示在活动源代码窗口中的源文件。与[Edit]菜单中的[Edit Source]功能相同。
Ctrl+G	执行一个搜索。打开与当前窗口相对应的搜索对话框。与 [View]菜单中的[Search…]功能相同。
Ctrl+J	移动显示位置。打开与当前窗口相对应的每一个对话框。与 [View]菜单中的[Move…]功能相同。
Ctrl+M	从跳转目标地址中显示内存内容，该地址由当前窗口中被选中的数据指定。打开存储器窗口。与 [Jump]菜单中的[Memory…]功能相同。
Ctrl+O	装载一个查看文件，源文件或者文本文件。 打开查看文件加载对话框。根据文件名后缀的不同该操作将有所不同。 查看文件: 在相应的窗口中显示该文件。 其它: 在源文件窗口中显示该文件。 与 [File]菜单中的[Open…]功能相同。
Ctrl+S	将显示在当前窗口中的数据保存到查看文件。 与 [View]菜单中的[Save…]功能相同。
Ctrl+U	显示相应的源本文和源代码行，使用当前窗口中选定的数值作为跳转地址。打开源文件窗口。与 [Jump]菜单中的[Source Text]功能相同。
Ctrl+V	从剪贴板缓冲区中将内容粘贴到文本光标位置。
Ctrl+W	临时显示指定数据的内容。打开快速观察对话框。 与 [View]菜单中的[Quick Watch…]功能相同。
Ctrl+X	将选定的字符串剪切，并存于剪切板缓冲区中。与 [Edit]菜单中的[Cut]功能相同。
Ctrl+Shift+Left arrow key	将选择范围向左边扩展到一个字。
Ctrl+Shift+Right arrow key	将选择范围向右边扩展到一个字。

附录 D　消息

D.1　显示格式

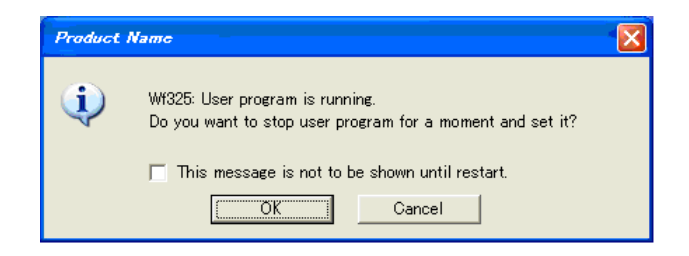
消息被输出到错误/ 警告对话框中。

当错误对话框弹出后，通过按 F1 键相关的在线帮助文件将会显示出来。

如果在显示的对话框中的[This message is not to be shown until restart]复选框被选择，这个消息将不再显示，直到 ID850QB不再运行。 当 ID850QB 下次被启动的时候，这个消息能被显示 (默认值: Cleared).

图 D-1 错误/告警对话框





D.2 消息类型

ID850QB 输出下列类型的消息。

表 D-1 消息类型

类型	意义
A	中止错误 停止处理，并且结束调试器。 如果这个错误发生，则调试将不能继续进行。
F	失败 停止处理，并且打开的窗口和对话框都被关闭。
W	警告 停止处理，但是打开的窗口和对话框都没被关闭

D.3　消息列表

< From X0000 >　< From X1000 >　< From X2000 >　< From X3000 >　< From X4000 >　< From X5000 >
< From X6000 >　< From X7000 >　< From X8000 >　< From X9000 >　< From Xa000 >　< From Xb000 >
< From Xc000 >　<From Xd000 >　< From Xe000 >　< From Xf000 >

(1)　From *X*0000

F0002:　这个特性没被支持。

F0100:　不能与ICE进行通讯。 请确认PC机接口板的设备驱动是否正确安装。

1) 检查电源及电缆的连接。

2) 驱动可能没有正确安装。 请重新安装驱动。

A0101:　不能找到初始化文件 (expc.ini)

A0102:　找不到主机。

F0103:　传输给ICE的数据超时。请确认ICE的电源，接口电缆的连接，或者PC接口板的I/O地址。

F0104:　从ICE获取数据超时。请确认ICE的电源，接口电缆的连接，或者PC接口板的I/O地址。

A0105:　读取设备文件失败。 (*file name*)

1) 必需的文件可能损坏。 重新安装设备文件。

A0106:　接收的数据不合法。

1)检查在线仿真器的电源，电缆连接以及接口板的设置并且重新启动调试器。

A0107:　不能与ICE通讯。

A0108:　读取初始化文件失败 (expc.ini)。

A0109:　不能与ICE通讯。 请关闭调试器然后检查ICE的电源或电缆连接，并复位调试器。

1) 在USB通讯过程中可能产生了一个错误(比如电源或电缆断开)或IECUBE出问题。**[IECUBE]**

A010a:　不能同时运行调试器和实例。

1) QB-Programmer, MINICUBE2 自检测工具 or OCD 检测器正在运行，请先关闭。

2) MINICUBE2 正在被　ID78K0-QB ，ID78K0S-QB 或 ID78K0R-QB 使用，请先关闭。

3) MINICUBE2 正在被　MULTI 或 C-SPY使用，请先关闭。

A01a0: 没有来自 evachip 的响应。 请确认CLOCK 或 RESET WAIT, HLDRQ 等信号。 没有来自 CPU 的响应。 请确认CLOCK or RESET WAIT, HLDRQ 等信号。 1) 检查HOLD信号，WAIT信号，CLOCK信号等，IOR 数值 (或 SFR 数值)可能是不正确的。
A01a1: 读取 ie703000.ie时失败。
A01a2: 断点板未连接。
A01a3: 仿真板未连接。
A01a4: ICE板配置不一致。
A01a5: POD/EM1板未连接。
A01a6: 执行器正在运行。 1) MINICUBE2 正在被 MULTI 或 C-SPY使用，请先关闭。
A01a8: 没有找到配置文件 (lv8hw.ini)
A01ad: 请更新PC接口板的设备驱动。 1) 驱动程序可能版本太旧。 安装最新的驱动程序。
A01ae: 读取配置文件时失败 (lv8hw.ini) 。
A01af: 在运行监视器指令时失败。
A01b0: 不能与监视程序通讯。 请检查可用的通讯端口，CPU板的设置或电缆的类型。
A01b1: 不能与监视程序通讯。 请关闭调试器，并检查CPU板电源或电缆的连接然后重启调试器。
A01b2: 仿真器的固件是旧版本。 请用最新的固件更新。 1) 使用 MQB2UTL更新软件。
F0200: 发生校验错误，不能写入内存。 1) 外部存储器不能被访问，可能是没有被设置。下载之前使用IOR窗口或者挂钩程序改变访问外部存储器必须的寄存器值。
F02a0: 总线保持错误。 1) CPU处在总线保持状态。请复位调试器。
F02a2: 不能强制产生断点。
F02a3: 持续复位。 1) reset释放后，当MINICUBE2 的RESET管脚没变成高电平时出现这个类型的错误。 下面是假设的原因。[MINICUBE2] - 与目标系统相连接时出现问题。(RESET).
F02d2: 没有足够的用于trace-buffer的内存。
F0300: 用户程序正在运行。
F0301: 用户程序正在断点期间。
F0302: 用户程序正在被跟踪。

F0303:　没有被跟踪。
F0304:　跟踪内存没被设置。
F0306:　不存在跟踪区块。
F0307:　不存在事件条件。
F0308:　未进行定时器测量。
F0309:　不存在触发结构。
F030a:　跟踪处于停止状态。
F030b:　指定的捕捉事件没有登录。
F030c:　指定的存根事件没有登录。
F030d:　定时器正在运行。
F030e:　内存复制区域重叠。
F030f:　跟踪已经被设定。
F0310:　事件条件没被设定。
F0311:　太多有效的定时器事件条件。
F0312:　指定的定时器事件没被设定。
F0313:　非法的映像范围。 1) 在配置对话框中检测映射范围。 当映射到外部存储器后，在下载之前，使用IOR窗口或者挂钩程序，改变访问外部存储器必须的寄存器的值。
F0314:　只有跟踪延迟模式可以由延迟触发器设置。
F0315:　没有跟踪延迟模式，不能设置延迟触发器。
F0316:　映射的数目溢出。
F03a0:　目标没被开启。 1) 检查目标系统电源。 检查在线仿真器和目标板的电缆连接。 检查确保VDD 信号被输入到目标板的连接端子。
F03a1:　正在进行单步执行。
F03a2:　正在运行定时器和跟踪器。
F03a3:　混合使用事件关联和BRS事件。
F03d0:　反跟踪正在运行。
F03d1:　反-跟踪处于停止状态。
F03d2:　反跟踪运行点超出了最早的框架。
F03d3:　除事件关联的 Phase1外，寄存器状态或存储器状态不能被设置。
F03d4:　不存在反跟踪信息。
F03d5:　最后的指令不能往后单步执行。
F0400:　非法的条件。 1) 使用的在线仿真器设置与配置对话框中的设置不匹配。 检查芯片选择。

F0401:　定时器测量结果溢出。

F0402:　路径数有太多事件条件。

F0403:　太多地址范围条件。

F0404:　太许多同步可用事件条件。

F0405:　太多捕捉事件。

F0406:　太多存根事件。

F0407:　太多初始化数据。

F0408:　太长的搜索数据 (>16 bit).

F0409:　太长的搜索数据.(>查找范围)

F040a:　太多关联事件条件。

F04a0:　软件断点条件数目溢出。

F04a1:　没有足够的供给仿真的内存。

F04a2:　总线大小太多分区。

F04a3:　太多运行事件条件。

F04a4:　太多总线事件条件。

A0600:　缓冲区中的内存不足。

1) 没有足够的系统内存。关闭打开的文件和正在运行的应用程序。

A0601:　操作系统没有足够的资源。

F0b20: 事件号不能被使用.

F0b61:　区段跟踪事件条件溢出。

F0b66:　在执行事件之前不能同时使用断点和软件中断。

1)　这是因为在运行前的一个断点被用于实现一个软件断点。 [MINICUBE]

F0b80:　硬件错误复位。

F0c00:　监视文件读取错误。

1) 必须的文件可能被损坏。 请重新安装调试器。

A0c01:　在访问寄存器的时候，CPU处理超时。

1) 检查时钟信号等，寄存器数值可能不正确。

A0c02:　在访问存储区的时候，CPU处理超时。

1) 检查HOLD、WAIT、CLOCK等信号，存储区中的数值可能不正确。

A0c03:　在访问I/O寄存器的时候，CPU处理超时。

1) 检查HOLD、WAIT、CLOCK等信号，I/O寄存器的数值可能不正确。

F0c04:　没有找到外部闪存数据库文件。

F0c20:　保护区不能被访问。

F0c21:　内存处于未准备状态。

W0c22: 内存未准备状态被取消。

F0c23:　总线持续占用。

1) 检查目标板的设置, 或者屏蔽HOLD管脚。

F0c24:　不能切换到调试模式。 1) 检查时钟信号。 可能由一个被停止的时钟或一个慢的时钟所引起。 2) 当监视程序在复位释放后得不到不回应时，会发生这个错误。 下面是假定的原因。 **[MINICUBE2]** - 连接目标系统时出现问题 (FLMD0)
F0c25:　闪存宏服务ROM被访问或者单步执行。 1) 请执行 [Go]运行或CPU复位。
F0c26:　FLMD 终端处于写保护的状态中。 1) FLMD 不在写-允许状态。 检查 FLMD0 和 FLMD1 管脚的状态。
F0c27:　安全标志在写保护的状态。 1) 闪存的安全标志不允许写入、块擦除或者芯片擦除。任何数据不能写入闪存。
F0c28:　内部RAM不足,闪存写入未执行。 1) 内部RAM大小不足4 KB而且闪存自编程不能运行。
F0c29:　闪存的空白检测失败。
F0c2a:　闪存的擦除失败。
F0c2b:　闪存的写入失败。
F0c2c:　闪存的内部校验失败。
F0c2d:　写入闪存时失败。
F0c2e:　没有来自闪存宏服务的响应。
F0c2f :　来自闪存宏服务的响应不正确。
F0c30:　闪存I/O寄存器操作禁止设置需要被取消。
F0c31:　在持续状态下的STOP模式。 不能强制产生断点。 请释放STOP模式或复位CPU。
F0c32:　请在单一芯片模式0下写入闪存。
F0c33:片上调试功能不能禁止。 1)　企图将"0"写入到ID码的MSB中 不可将0写入到MSB;否则不能执行片上调试功能。 **[MINICUBE2]**
F0c34:　写入片上调试器保留区被禁止。 1) 企图写入一个非法值到片上调试保留区。不可写入保留区域。 [MINICUBE2] 2) 程序覆盖监视程序上的一个分支代码到复位入口代码时失败。将复位入口代码转换为指定的代码。 [MINICUBE2]
F0c35:　不正常的内部ROM大小。这个大小不同于设备的默认值。
F0c36:　不正常的内部ROM大小。这个大小不同于设备的默认值。
F0c37:　闪存写入时的电压太低。
F0c38:　扩展监视区域不为空。

F0c39: 实时RAM监视失败。
F0c3a: 不支持写数据闪存区域。
F0c3b: 不能把数据写入闪存，因为不在数据闪存的环境中。
F0c3c : 外部闪存信息没有设定。
F0c3d: 不能擦除外部闪存。
F0c3e: 不能写入外部闪存。
F0c40: 不能改变有效事件条件状态。
F0c41: 覆盖率测试正在运行。
F0c42: 切换到调试模式时监听失败。请复位CPU。
F0c43: 不能连接仿真器。 1) 如果使用的是桌上型计算机并且两个以上PC卡插入到计算机中，切换设置可能出错。检查设置， 或者可能已经发生故障。 **[MINICUBE]** 2) 检查在线仿真器的电源和电缆连接。如果使用的是桌上型计算机并且两个以上PC卡插入到计算机中，切换设置可能出错。检查设置， 或者可能已经发生故障。 **[MINICUBE]** 3) 检查在线仿真器的电源和电缆连接。 **[MINICUBE2]**
F0c44: 覆盖率测试正在运行。
F0c45: 仿真器内部失电复位，无法运行程序。
F0c46: 在闪存自仿真过程中，无法改变的内部ROM、RAM、RAM监视器或者DMM。
F0c47: ROM校正仿真的已经被使能了。
F0c60: 除了断点条件之外，运行前的事件不能被设置。 1) 使用event-after-execution。
F0c61: 不能登录无法用于硬断点事件号。
F0c62: 保留的硬断点事件号不能被使用。
F0c63: 不能设定事件关联条件。
F0c64: 太多的ROM-emulation-RAM区域。
F0c67: 在区块操作的时候没有写入闪存。
F0c68: 不能用未定义的代码执行ROM校正仿真。
F0c69: 数据闪存区之外地址已经被指定。
F0c6a: 外部闪存区之外地址已经被指定。
F0c70: DCU不能被访问。 1) 设备文件选择可能不正确。在配置对话框中的芯片选择中选择一个支持目标芯片的设备文件。检查芯片的电源。 检查信号线的连接 (DCK ， DMS ， DDI ， DDO 和 DRSTZ)。 用N-Wire检测器上的DCK波型检测功能检测噪声水平。 **[MINICUBE]** 2) IE可能发生故障。 **[IECUBE]**

F0c71 不能复位 1) 检查时钟信号。 这可能由一个被停止的时钟或一个慢的时钟所引起。 2) 当复位后，MINICUBE2 的RESET管脚没成为低电平时，可能发生这个错误。下面是假设的原因。[MINICUBE2] -和目标系统的连接可能问题 (RESET)
F0c72 监视器内存不能被访问。 1) 在配置对话框中修改主要OSC值，并检查主时钟的设置，用N-Wire检测器上的DCK波型检测功能检测噪声水平。问题可能也由内部芯片问题引起。[MINICUBE] 2) 在配置对话框中修改主要OSC值。如果问题仍未解决，可能是IE出现故障。 [IECUBE] 3) 当写入监视程序失败的时候会发生这个错误。 下面是假设的原因。 -和目标系统的连接可能问题 (通讯脚或者FLMD0脚) - 在配置对话框中选择了错误的CSI或者UART。 - 输入到配置对话框中时钟区域的振动频率与目标系统上的器件频率不同。[MINICUBE2]
F0c73: 监视器不能运行。 1) 用N-Wire检测器上的DCK波型检测功能检测噪声水平。这个问题也可能由一个内部芯片问题所引起。**[MINICUBE]** 2) IE 可能发生故障。 [IECUBE] 3) 当ID代码在用户程序执行期间改变的时候发生错误。不能在用户程序执行期间改变ID代码。[MINICUBE2]
F0c74: CPU不能被访问寄存器。 1) 用N-Wire检测器上的DCK波型检测功能检测噪声水平。这个问题也可能由一个内部芯片问题所引起。**[MINICUBE]** 2) 设备文件选择不正确。在**配置对话框**中的芯片选择中选择一个支持目标芯片的设备文件。如果这样不能解决问题,，IE 可能发生故障。 [IECUBE] 3) [MINICUBE2]尝试使用不支持目标系统上的设备的设备文件启动调试器，检查调试器里的设备是否正确选择了。
F0c75 切换到调试模式时监视失败。请请复位CPU。
F0c76: DCU 访问开始的时候初始状态不正常。 1) DCU 访问开始的时候初始状态是不正常的。 (没有启动和仍处于复位状态)。检查信号线的连接 (DCK ，DMS ， DDI ， DDO 和 DRSTZ) 。用N-Wire检测器上的DCK波型检测功能检测噪声水平。 [MINICUBE] 2) 设备文件选择可能不正确。在配置对话框中的芯片选择中选择一个支持目标芯片的设备文件。如果这样不能解决问题，IE 可能发生故障。 **[IECUBE]**

F0c77:　DCU 访问不正常。 1) DCU 访问不正常(校验错误) 。检查信号线的连接 (DCK， DMS， DDI， DDO 和 DRSTZ) 。用N-Wire检测器上的DCK波型检测功能检测噪声水平。[MINICUBE] 2) IE 可能发生故障。 [IECUBE]
F0c78:　在读取跟踪数据时失败。
F0ca0:　不能与ICE通讯。 请检查ICE电源，接口电缆的连接或者PC接口板上的I/O地址。
F0ca1:　监视文件没被找到。 1) 必需的文件可能被损坏。 重新安装调试器。 2) 当一个不被支持的设备被选择的时候，会发生这个错误，检查是否选择了支持类型的设备。
F0ca2:　这个设备文件不包含片上调试信息。 1) 企图启用不支持片上调试功能的设备文件。设备文件版本可能太旧，安装最新的设备文件。[MINICUBE][MINICUBE2] 2) IE 可能发生故障。 [IECUBE]
F0ca3: 在设备文件里面包含了不支持的片上调试信息中。 1) 一个不明标志包含在设备文件的片上调试信息中。执行模块可能版本太旧，安装最新的执行模块。
F0ca4 设备文件没有包括 IECUBE 信息。 1) 企图启动设备文件不支持的IECUBE。设备文件可能版本太旧，安装最新的设备文件。
F0caf:　跟踪区块不能被跳过。

(2)　From *X*1000

A1000:　不能初始化ICE。
A1001:　指定型号的入口不存在。
A1002:　不能重新分配内部RAM。
F1003:　非法的重新分配地址。
F1004:　非法的条件。
A1005:　无效的属性。
F1006:　非法的地址。
A1007:　在ICE没有足够的内存。
A1008:　没有足够的内存分配给表格。 1) 没有充足的系统内存。 关闭正在运行的应用程序和打开的文件。
A1009:　已经初始化。
A100a:　没被初始化。
F100b:　用户程序正在运行。

F100c: 已经指定了不同的总线大小。
F100d: 太大的总线大小。
F100e: 太大的总线分区。
W100f: 对象没被打开。
F1010: 非法的映像范围
F1011: 设定内部的ROM和RAM失败
F1012: 不被支持的特性。
F1013: 没有终端名称。
W1014: 数据不存在。
A1015: 可编程的-IOR 不存在
F1016: 可编程的- IOR 是不可移动的。 1) 必需的文件可能被损坏。 重新安装最新的设备文件。
F1017: I/O保护映射是可能只是一个目标属性。
F1018: 非法的内部ROM大小。
F1019: 非法的内部ROM大小或内部的RAM大小。
F101d: 数据闪存区不可移动。
F101e: 数据闪存区不存在
A10ff: 不能与ICE通讯。
A1dbe: 调试器内部发生错误。

(3) From *X*2000

F2000: 非法的 IOR 名称。
A2001: 非法的地址。
F2002: 用户程序正在运行。
F2003: 非法的 IOR 号
F2004: 非法的bit数。
W2005: IOR 读取保护属性被指定。
F2006: 隐藏的 IOR 被指定。
F2007: 禁止读写的IOR 被指定。
F2008: 指定的IOR不存在。
A2009: 设备文件是被损坏的，或者文件内部错误。
F200a: 为IOR指定了非法的数值。
A200b: 不能复制
A200c: 内存不足。 1) 没有充足的系统存储器。关闭正在运行的应用程序和打开的文件。
W200d: 没有为 IOR 设定数据初值

F200e: IOR 区域不能被访问。
A20ff: 不能与ICE通讯。
A2222: 非法的条件。

(4) From *X*3000

F3000: 没有映射地址被存取。 1) 程序的分配地址和调试器的地址不可能相配。 在配置对话框根据编辑中的连接指示文件中所指定分配地址设置外部存储区的映射。当外部存储区映射已经被运行的时候,在下载前，使用IOR 窗口或挂钩程序修改访问外部存储器所必需的寄存器值。
001: 存储器有不同的数值。
F3002: 非法的开始地址。
F3003: 非法的结束地址
F3004: 非法的开始地址和结束地址
F3005: 非法的条件。
F3006: 用户程序正在运行。
F3007: 验证错误。
F3008: 没有指定条件。
F3009: 参数大小排列和访问大小排列不一致
F300a: 指定的地址排列和访问大小排列不一致
F300b: 源地址排列和访问大小排列不一致
F300c: 目标地址排列和访问大小排列不一致
F300d: 非法的结束地址。
F300e: 在指定的区域中的不同访问大小。
F300f: 在源和目的地区域不同的访问大小
F3010: 在目的区域中不同的访问大小。
F3011: 不同的访问大小，源 &目的区域。
A3012: 内存不足。 1) 没有充足的系统存储器。关闭正在运行的应用程序和打开的文件。
F3013: 写入 DMM 时失败。
F3014: 映射区溢出。
F3015: 处理被中断。
F3016: 不被支持的特性。
A30ff: 不能与ICE通讯

(5) From *X*4000

F4000: 不能删除指定的事件。

1) 由于在另一个条件中被使用，指定的事件不能被删除。在删除前,关闭此事件其他应用。

F4001: 非法的表单数。

F4002: 非法的开始地址。

F4003: 非法的结束地址。

F4004: 非法的状态。

F4005: 非法的数据。

F4006: 指定的事件号已经被使用。

F4007: 太多相同的事件被登录

F4008: 指定的事件没有被登录。

F4009: 非法的数据大小。

F400a: 非法的模式。

F400b: 设置值不准确。

F400c: 事件关联条件不能够作为区段跟踪条件。

F400d: 登录了太多相同事件。 (>=32767)

F400e: 指定的事件条件不存在。

F400f: 非法的事件关联条件。

F4010: 函数没找到。

A4011: 内存不足。
1) 没有足够的系统存储器。关闭正在运行的应用程序和打开的文件。

F4012: 定时器无效。

W4013: 存取大小与映射总线大小不同。

F4014: 不能使用软件断点。

F4015: 不能使用事件条件指定地址范围的

F4016: 不能改变事件条件。

F4017: 在奇数地址中不能访问字变量。

A4018: 内存不足。
1) 没有足够的系统存储器。关闭正在运行的应用程序和打开的文件。

F4019: 不支持该特性。

F401a: 没有事件。

F401b: 不能使用标签事件

W401c: 该区域不能设置软件中断。

F401d: 定时器的启动与结束事件设置不一样。

F401e: 太多跟踪事件。

F4401f: 路径计数不能被建立。
F4020: 在事件执行之前,地址范围不能被建立。
F4021: 事件条件号溢出。
F4022: 软件 DMM 条件号溢出。
F4023: 实时调用条件号溢出
F4024: 软件断点调用条件号溢出
F4025: 非法的捕捉条件。
F4026: 太多事件条件不能被设定为事件关联条件Phase1和Phase2。
F4027: 可以被设置为内部ROM的软件断点条件号溢出。
F4318: 非法的内存区设定。

(6) From *X*5000

A5000: 非法的设备文件类型。
A5001: 内存不足。 1) 没有足够的系统存储器。关闭正在运行的应用程序和打开的文件。
A5002: 不能打开设备文件。
A5003: 设备文件的读错误
A5004: 不能关闭设备文件。
A5005: 非法的设备文件格式。 1) 必需的文件可能被损坏。 重新安装设备文件。
A5006: ICE初始化失败。
A5007: 设备文件已经中断，或者文件中有错误。
F5008: 不能打开设备文件。 1) 必需的文件可能被损坏。 重新安装设备文件
F5009: 不能开启 ie703000.ie
F500a: 指定的设备文件是非法的版本。 1) 必需的文件可能被损坏。 重新安装设备文件。
W500b: 指定的设备文件不能重新分配内部RAM。
A500c: 读取expc.ini时失败。
A500d: 内存不足。 1) 没有充足的系统存储器。关闭正在运行的应用程序和打开的文件。
W500e: 没有将要指向设备文件的标签数据。
A5300: 非法的设备文件类型。
A5301: 内存不足。 1) 没有充足的系统存储器。关闭正在运行的应用程序和打开的文件。

A5302: 不能开启数据库文件。 1) 必需的文件可能被损坏。 重新安装调试器和设备文件。
A5303: 读取数据库文件出错。
A5304: 不能关闭数据库文件。
A5305: 非法的数据库文件格式。 1) 必需的文件可能被损坏。 重新安装调试器和设备文件。
A5306: 数据库信息已经被初始化。
A5307: 数据库信息不存在。
F5308: 不能打开指定的数据库文件。 1) 必需的文件可能被损坏。 重新安装调试器。
F5309: 指定的数据库文件是非法的版本。 1) 必需的文件可能被损坏。 重新安装调试器和设备文件。

(7) From *X*6000

F6000: 当前功能不存在。
F6001: 非法的变量名。
F6002: 非法的条件。
F6003: 非法的函数名。
F6004: 输出缓冲区大小溢出。
F6005: 非法表达式。

(8) From *X*7000

F7000: 非法的模式。
F7001: 用户程序正在运行。
F7002: 用户程序已经停止。
F7003: 跟踪允许。
F7004: 跟踪内存没被设置。
F7005: 函数返回地址不存在,不能单步运行。
W7010: 源信息不存在。
W7011: 单步运行的未知结果。
A7012: 内存不足。 1) 没有充足的系统存储器。关闭正在运行的应用程序和打开的文件。
A70fe: 总线保持错误。 1) CPU处在总线保持状态。 复位调试器。
A70ff: 不能与ICE通讯。
F7801: 单步运行的结束等候状态被取消。

F7802:　单步运行的结束等候状态被取消。
F7f00:　单步运行被中止。
F7f02:　单步运行被挂起。
A7f03:　取消RUN/STEP时失败。
F7f04:　不能运行非映射区域。
F7f05:　不支持该特征。

(9)　From *X*8000

F8000:　指定的文件没找到。
F8001:　非法的行号。
F8002:　当前的信息没被设定
F8003:　非法的地址。
F8004:　不支持该特性。

(10)　From *X*9000

A9000:　指定的寄存器符号不存在。
A9001:　指定的寄存器符号ID不存在
F9002:　非法的数值。
A9003:　非法的条件。
A9004:　寄存器大小太大。
F9005:　不支持该特征。

(11)　From *X*a000

Fa001:　非法表达式。
Fa002:　开始地址大于结束地址。
Fa003:　非法的源路径。
Fa004:　太长的表达式。
Aa005:　内存不足。 1) 没有充足的系统存储器。关闭正在运行的应用程序和打开的文件。
Fa006:　非法的参数。
Fa007:　非法的程序号。
Fa008:　源路径没被设定。
Fa009:　文件没找到。
Fa00a:　不能打开文件。 1) 文件已损坏或者不存在。 重新创建文件
Aa00b:　不能关闭文件。

Aa00c: 读取文件时失败。 1) 文件已损坏或者不存在。 重新创建文件。
Fa00d: 加载模块中没有源文件。
Fa00e: 非法的行号。
Fa00f: 变量不存在。
Aa010: 不能与ICE通讯。
Fa011: 不能访问寄存器。
Fa012: 不能访问存储器。
Aa013: 读取文件错误。
Fa014: 试图打开二进制文件。
Fa015: 不能创建临时路径。 1) 磁盘已满。 删除或移动不必要的文件，增加磁盘的可用空间。
Fa016: 不能创建临时文件。 1) 磁盘已满。 删除或移动不必要的文件，增加磁盘的可用空间。
Fa017: 不能移除临时文件。
Fa020: 不支持该特性。
Fa021: 赋给寄存器的变量不能被指定。
Fa022: 字符不能够用作文件夹名，或者文件夹不存在。

(12) From *X*b000

Fb000: 非法的指令行。
Fb001: 在指定的加载模块文件中不存在程序信息。
Fb002: 文件没找到。
Fb003: 函数没找到。
Fb004: 选择的加载模块与已经加载的模块是不同类型 (芯片) 的。
Fb005: 变量没找到。 1) 地址无法找到。 指定一个含有地址信息的位置。
Fb008: 非法表达式。
Ab009: 内存不足。 1) 没有充足的系统存储器。关闭正在运行的应用程序和打开的文件。
Fb00a: 加载模块文件中有非法符号。
Fb00b: 当前程序不存在。
Fb00c: 当前文件不存在。
Ab00d: 当前函数不存在。
Ab00e: 当前行不存在。
Ab00f: 标签没找到。

Ab010:　载入符号表时失败。

Ab011:　 非法的行号。

Fb012:　太大的行号。

Ab015:　读取文件错误。
1) 文件被损坏或者不存在。 重新创建文件。

Ab016: 不能打开文件。
1) 文件被损坏或者不存在。 重新创建文件。

Ab017：　写入文件错误。
1) 文件被损坏或者不存在。 重新创建文件。

Ab019:　读取文件错误。

Ab01a:　不能关闭文件。

Fb01b:　太长的加载模块文件名。

Ab01c:　太多的任务类型入口。

Fb01d:　地址没找到。

Wb01e: 没有调试信息.(没在调试编译模式下编译)

Fb01f:　 不能找到结构成员。

Fb020:　不能找数值。

Fb021:　在加载模块文件中，既没有调试信息也没有加载模块文件的符号信息。
1) 为了用附加的调试信息产生一个加载模块,使用调试生成模式进行生成。

Fb022:　非法的行号。

Ab023:　当前堆栈结构不是激活的。

Ab024:　不同的区段。

Fb026: 太多维数组(>4)

Fb027:　发现文件终点。
1) 指定的文件可能被损坏。 重新创建文件。

Fb028:　不支持该特性。

Fb029:　非法的地址。

Ab02a:　不能与ICE.通讯。

Fb02b:　不能将当前的追踪PC值压入堆栈。

Fb02c:　 一个函数中有太多区块。

Fb02d:　非法的参数。

Fb02e　文件在源路径中不存在。
1) 一停止程序，调试器要显示的源无法找到。检查调试选项对话框中目录是否连接到源路径，或者检查源是否与输出文件在同一个目录里面。参考汇编窗口的错误信息显示，并且检查相应的连接路径。

Fb02f:　因为优化的原因，信息已经被删除。
Ab030:　监视器超时。 1)检查在线仿真器的电源，电缆连接和接口板的设置，并重启调试器。
Ab031:　已经在存储器中设定。
Ab032:　超出范围
Ab033:　LP没被储存。
Fb034:　不能从当前执行的PC位置返回。
Wb036: 超出变量区域。
Fb037:　太多行号信息。
Fb038:　编译器版本不匹配。 1) 用最新的编译器重新创建模块。
Ab039:　在载入调试信息时失败。
Ab03a:　没有更多的区段信息。
Fb040:　指定的文件不是加载模块。 1) 这不是一个链接输出文件。 在从链接器输出的时候，源调试不能和装载模块一起执行。指定从链接器输出的加载模块。
Ab041:　加载模块中有太多文件下载。
Wb042: 符号模块没被初始化。
Fb32e:　非法的端口数。
Fb32f:　非法的端口名称。
Fb330:　非法的端口位置。
Fb331:　非法的增量数。
Fb332:　内存区域端口没被设定。
Fb333:　非法的插槽数。
Fb334:　内存区域没被设定。
Wb335: 太长的符号名。

(13)　从　*X*c000

Fc001:　不能打开文件。 1) 文件被损坏或者不存在。 重新创建文件。
Ac002:　不能关闭文件。
Ac003: 读取文件错误。 1) 文件被损坏或者不存在。 重新创建文件。
Ac004:　读取文件错误。
Fc005:　非法的文件类型。
Fc006:　加载模块的类型 (芯片) 是非法的。

Fc007:　指定的文件不是加载模块。 1) 这不是一个链接输出文件。 在从链接器输出的时候，源调试不能和加载模块一起执行。指定从链接器输出的加载模块。
Fc008:　指定的加载模块文件 (EFL) 是旧版本。
Ac009:　内存不足。 1) 没有充足的系统存储器。关闭正在运行的应用程序和打开的文件。
Fc00a:　没有访问映射地址。
Fc00b:　加载模块没被载入。
Fc00c:　非法的参数。
Fc00d:　用户程序正在运行。
Fc00e:　用户程序正在被跟踪调试。
Fc00f:　中断。
Ac010:　不能与ICE通讯
Fc011:　非法的加载模块文件格式。
Fc012:　校验和错误。
Fc013:　上传太宽的地址范围(>1 M bit) 。
Fc014:　写入文件错误。 1) 文件被损坏或者不存在。 重新创建文件。
Fc015:　非法的程序号。
Fc016:　装载信息已满。
Wc017: 符号信息重复,请复位变量。
Fc018:　指定的文件不是加载模块。 1) 这不是一个链接输出文件。 在从链接器输出的时候，源调试不能和加载模块一起执行。指定从链接器输出的加载模块。
Fc019:　写入存储器时失败。
Wc01a: BSS 区域被赋值给非映射区域。 1) 当程序被运行的时候,一个非映像断点可能发生。 或者通过一个链接指示分配BSS区域到内部RAM，或通过调试器的配置对话框映射仿真内存或目标内存到BSS区域。
Fc01b:　可编程 IOR 地址没被指定。 1) 必需的文件可能被损坏。 重新安装调试器。
Wc01c: 可编程 IOR 地址不匹配。 1) 必需的文件可能被损坏。 重新安装调试器。
Wc01d: 被载入的类型 (芯片)与被选择的加载模块不相同。
Fc01e:　不支持闪存擦除。
Fc100:　不支持该特性。

(14)　From *X*d000

Ad000:　调试器内部发生错误。
Ad001:　内存不足。 1) 没有充足的系统存储器。关闭正在运行的应用程序和打开的文件。
Ad002:　读取初始化文件 (expc.ini) 失败。
Ad003:　ICE未连接。
Fd004:　不能找到动态链接库。

(15)　From *X*e000

Fe000:　非法的参数。
Fe001:　非法的开始地址。
Fe002:　非法的结束地址。
Fe003:　太大的尺寸。
Fe004:　不能打开文件。 1) 文件被损坏或者不存在。 重新创建文件
Fe005:　读取文件时失败。 1) 文件被损坏或者不存在。 重新创建文件。
Fe006:　读取文件错误。
Fe007:　写入文件错误。 1) 文件被损坏或者不存在。 重新创建文件。
Ae008:　内存不足。 1) 没有充足的系统存储器。关闭正在运行的应用程序和打开的文件。
Fe009:　非法的文件格式。
Fe00a:　校验错误。
Fe010:　不支持该特征。

(16)　From *X*f000

Af000:　内存不足。 1) 没有充足的系统存储器。关闭正在运行的应用程序和打开的文件。
Ff000:　内存不足。
Ff001:　没找到[XXX]。
Wf002:　没找到 [XXX] 。 是否从头搜索?
Wf003:　已经超过搜索区域。
Ff004:　遗漏参数。
Ff005:　非法的函数名。
Ff006:　非法的数目。

Ff007:	开始地址比结束地址大。
Ff008:	非法的变量或表达式。
Ff009:	[XXX] 这个文件是非法的类型。
Ff100:	磁盘不能够写入或已经写满。
Ff101:	文件没找到。
Ff102:	没有创建文件。
Ff103:	旧的工程文件版本。
Ff104:	非法的工程文件格式。
Ff105:	这是一个[XXX]工程文件。请选择一个正确的文件。
Wf106:	工程文件的CPU被改变。 必须退出调试器以得到新的CPU。 你想退出调试器吗?
Wf107:	工程文件的CPU被改变。 你想用这个CPU启动调试器吗?
Wf108:	已经打开了一个选中的与芯片 [XXX]不同的工程文件[YYY]。 它打开了吗？虽然芯片不能被改变?
Wf109:	工程管理器不能被用于这个这个版本上的调试器。 请使用 PM+.
Wf200:	没有遇到任何不同。
Ff201:	存储器映射错误。
Ff202:	验证错误。 1) 外部存储器无法被访问,因为它没被设定。 下载之前，使用IOR窗口或者Hook窗口改变必要的寄存器值以访问外部存储器。
Wf203:	当一个程序正在运行的时候, 重写内存时，程序执运行暂停。 你想重写一个存储器吗?
Wf204:	有一个被使能的软件断点,在 DMM 中失败。
Wf300:	你想把在 [XXX] 中的改动存档吗?
Ff301:	在事件条件上被用的符号不能被求值。
Wf302:	删除: [XXX]
Wf303:	[XXX] 被编辑。 删除: [YYY]?
Wf304:	[XXX] 被编辑。 保存: [YYY]?
Wf305:	[XXX] 已经存在。替换它吗?
Ff306:	这个名称太长。
Ff307:	在其他的类型有相同的名称。
Ff308:	一个不能被省略的地址。
Ff309:	非法的地址屏蔽。
Ff30a:	非法的数据屏蔽。
Ff30b:	非法的 ext 探查屏蔽。
Ff30c:	非法的 ext 探查数据。
Ff30d:	非法的执行计数

Ff30e: 非法的寄存器命名。

Ff30f: 非法的寄存器

Ff310: 非法的延迟计数。

Wf311: 只有一 [XXX] 能被使能。 你想使能 [YYY] 吗

Ff312: [XXX] 已经存在。

Ff313: 事件号已经存在。

Ff314: 事件名称没被设定。

Ff315: [XXX] 已经存在。

Ff316: 超出最大[XXX]使能的事件数。请关闭其它已经使能的事件[YYY]

Ff317: 超出了最大设置事件的数目。

Ff31e: 非法的开始地址。

Ff31f: 非法的结束地址。

Ff322: 非法的计数率。

Ff323: 非法的超时断点计数。

Ff324: 同时可以指定区段和限制。

Wf325: 用户程序正在运行。 你想要暂停用户程序，并且设定它吗?

Wf326: 用户程序正在运行。你想要暂停用户程序，并且删除它吗?

Ff327: RAM监视器被使能,一个软件断点不能被使用。

Ff350: 有一个不在中间的事件相位。

Ff351: 相同的事件被包含在链接里，并未被使能。

Ff352: 一个事件没被指定。

Ff357: AND事件在相位中。

Ff400: 范围映射错误。

Wf401: 清除范围?

Ff500: 非法的符号。

Ff501: 非法的数值。

Ff502: 非法的参数。

Ff503: 超过了最大符号数目。

Ff504: 这个变量不能被设定为一个断点。

1) 断点不能被设定为如下的变量。

- 局部变量,静态变量
- 数组变量，结构体/共用体的成员变量
- 寄存器/ 周边I/O寄存器
- 变量表达式

Wf600: 保存工程文件?

Wf601: 当连接目标系统的时候,请打开目标系统。

1) 当一个目标系统没被连接的时候，只要点击 <OK>按钮。

Wf602: 请改变一个模式屏蔽条件或连接目标系统
Ff603: 不正确的ID码。 1) 这可能由下列原因造成： **[MINICUBE]** - ID码不正确。 -> 输入正确的ID码。 - 因为 FLMD0 管脚是高电平，所以内在的闪存处在写入模式中。 ->使 FLMD0 管脚成为低电平。 - 因为ID码 (地址 0 x79的第7位) 是 0 ，所以仿真器连接被设为禁止模式。 ->擦除内部闪存一次。
Af604: 不正确的ID码。 中止调试器。 1) 这可能由下列原因造成： **[MINICUBE]** - ID码不正确。 -> 输入正确的ID码。 - 因为 FLMD0 管脚是高电平，所以内部的闪存处在写入模式中。 -> FLMD0 管脚置为低电平。 - 因为ID码 (地址 0 x79的第7位) 是 0 ，所以仿真器连接被设为禁止模式。 ->擦除内在闪存一次。
Ff605: 请检查与目标板的连接。 1) 检查目标连接器 (TC) 的连接。 如果一个目标系统没被连接，请重新察看配置对话框中的目标设定。
Ff606: 请检查与目标板的连接，并且将电源打开。 1) 检查目标系统电源供应。 如果一个目标系统没被连接，请重新察看配置对话框的目标设定。
Wf607: 请检查交换适配器的连接。 1) 检查交换适配器(EA) 的连接。 如果目标系统没被连接，推荐打磨交换适配器
Ff608: 请拆开目标板。 1)电流可能从内部电源流到目标系统。 将目标系统连接器 (TC) 与转换器 (EA) 分开。如果目标系统没被连接，请重新察看配置对话框的设定。
Ff609: 请关闭目标板电源拆开它。
Af60a: 不正确的ID码。闪存被擦除，中止调试器。 1) 当擦除闪存内容的模式被设置时，如果ID认证地址0 x84导致失败，这个消息就会被显示。并且内存中的内容将被擦除。 [MINICUBE]
Af60b: 无效的 ID码。闪存被擦除，中止调试器。 1) 当在关闭电源仿真模式下使用调试器，调试器在目标系统电源断开后被激活，或者当闪存内容被清除时，如果目标系统连接电缆被断开，这个消息就会被显示。[MINICUBE]

Af60c:　在断点对象没被打开的时候。

Wf60d:　因为源路径已经超出4095个字符,回到原来的长度。

Wf700:　你想要下载加载模块文件吗?

Wf701:　你只想载入变量信息吗?

Wf800:　内存区域没被设定。

Wf801:　BANK地址必须在对象存储器中。

Ff802:　所有的事件被删除。 因为外部探针的使用被改变。

Ff803:　在当前配置上，这个事件地址是无效的

Ff804:　无效的PC值。

Ff805:　不能在这个地址上设置临时中断。

Ff806:　外部数据正在被调试器使用。

Ff900:　非法的I/O端口名。

Ff901:　存储器映射错误。

1) 指定的地址不合法。 检查能在增加I/O端口对话框中被指定的地址。

Ff902:　非法的访问大小。

Ff903:　非法的访问状态。

Ff904:　有相同的名称。

Wf905:　[XXX] 已经存在。 你想替换它吗?

Wf906:　你想登录在[XXX]里面的修改吗?

Ffa00:　没在PC位置上找到当前程序的[XXX]函数。

1)　指定的变量在main () 标识中:在调试器选择项对话框中可能找到。　在程序中设置一个主程序变量，默认值是_main。

Ffa01:　PC位置上的行信息没有找到

1) 当程序被停止的时，与程序计数(PC)值有关的源文件没被找到。下面是可能的原因之一：

- 源文件存在一个源路径没有连接到的地方。

-程序停止在源文件不存在的地方，比如库文件或者RX。

-程序循环，跳转到没被程序使用的地址，并且停止在那里。

Wfb00:　用户程序正在运行。 你想要阻止用户程序吗?

1)　<Yes> 按钮被选择, 用户程序的运行被停止，然后显示退出调试器对话框。如果它在调试选项窗口中被指定不显示退出调试对话框，那么ID850QB 被结束。

<No> 按钮被选择, 用户程序的运行没被停止，并且退出调试对话框不显示。ID850QB 没有结束。

Wfb01:　因为ID码的地址0 x79中的第7位是0, N-Wire仿真器从此被禁用。你要退出调试器吗？

Ffc00:	在线帮助窗口不能被启动。 请参考用户参考手册安装HTML帮助环境。
Ffd00:	指定[XXX]失败。
Ffe00:	超出RRM的最大范围。
Wfe01:	存在一个相同的RRM地址。
Wfe0b:	切换到闪存模式。除了当前事件外，它被全部清除了吗？没关系吗？
Ffe0c:	RAM监视器被使能。 当前软件断点完全被禁止，它没关系吗?
Fffff:	中断。

附录 E 索引

详细信息请联系：

（中国区）

网址：
http://www.cn.necel.com/
http://www.necel.com/

[北京]

日电电子（中国）有限公司
中国北京市海淀区知春路 27 号
量子芯座 7，8，9，15 层
电话：(+86)10-8235-1155
传真：(+86)10-8235-7679

[上海]

日电电子（中国）有限公司上海分公司
中国上海市浦东新区银城中路 200 号
中银大厦 2409-2412 和 2509-2510 室
电话：(+86)21-5888-5400
传真：(+86)21-5888-5230

上海恩益禧电子国际贸易有限公司
中国上海市浦东新区银城中路 200 号
中银大厦 2511-2512 室
电话：(+86)21-5888-5400
传真：(+86)21-5888-5230

[深圳]

日电电子（中国）有限公司深圳分公司
深圳市福田区益田路卓越时代广场大厦 39 楼
3901，3902，3909 室
电话：(+86)755-8282-9800
传真：(+86)755-8282-9899

[香港]

香港日电电子有限公司
香港九龙旺角太子道西 193 号新世纪广场
第 2 座 16 楼 1601-1613 室
电话： (+852)2886-9318
传真： (+852)2886-9022
2886-9044